

**SEMANTIC INTELLIGENCE FOR KNOWLEDGE-BASED
COMPLIANCE CHECKING OF UNDERGROUND UTILITIES**

by
Xin Xu

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Lyles School of Civil Engineering

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Hubo Cai, Chair

Lyles School of Civil Engineering, Purdue University

Dr. Jie Shan

Lyles School of Civil Engineering, Purdue University

Dr. Arif Ghafoor

School of Electrical and Computer Engineering, Purdue University

Dr. Nora El-Gohary

Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign

Approved by:

Dr. Dulcy Abraham

To my parents and my love

ACKNOWLEDGMENTS

I could not have completed this work on my own. There have been a great number of individuals who have supported me along the way on my Ph.D. journey and for whom I am very appreciative.

First of all, I would like to express my deepest appreciation to my advisor, Professor Hubo Cai, for the continuous support of my Ph.D. study at Purdue, for his patience, motivation, and immense knowledge. It is him that always encourages me to think outside the box and brings me into the wonderful world of research. I am very thankful for his patience in improving my academic writing skills. He spends a lot of time on revising my manuscripts and providing valuable suggestions. I enjoy talking with Professor Cai, which is also helpful to improve my logical and critical thinking skills. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D. study. The training I have received under his guidance has been invaluable and will be undoubtedly useful for the next chapter of my life.

I would also like to thank my committee of Professor Jie Shan, Professor Arif Ghafoor, and Professor Nora EI-Gohary for their constant support and guidance. I learned a lot from Professor Shan's GIS class and Professor Ghafoor's database class, which in turn helped me better understand my research problems and provided necessary technical solutions. I also learned a lot from Professor EI-Gohary, who is the leading researcher in the area of compliance checking. I am so grateful for the time they took to guide me to accomplish this work.

I would also like to acknowledge my current and previous lab mates – Shuai Li, Jaehyun Park, Chenxi Yuan, Jiannan Cai, Yuxi Zhang, JungHo Jeon, and Liu Yang. Shuai, Jaehyun, and Chenxi have set good examples in our lab, which encourage us to continue working hard to pursue our dreams. I am very lucky to have Jiannan as my lab mate. Her diligence and cleverness always “push” me to reflect myself and work hard in pursuit of excellence. I am also thankful to have Yuxi, JungHo, and Liu in our research team. They all have nice personalities and contribute a lot to creating a perfect working atmosphere in our lab. Every time when I am in our lab, I am happy.

I would also like to thank my wonderful friends here at Purdue University, especially the support of Xiaodong Jiang, Richard Wong, and Yingsheng Zhang. We have a weekly routine of

basketball game. I enjoyed every game with them and treasured the memory of winning the game together.

I would like to gratefully acknowledge the China Scholarship Council (CSC), National Science Foundation (NSF), and Joint Transportation Research Program (JTRP) that funded me during my Ph.D. study.

Finally, I would like to express my deepest love and appreciation to my parents for their constant and unwavering support during my life. Without their endless sacrifices and love, I would never have gotten as far as I have. Special thanks to Kaiwen Chen, my love, for all of the love and support that kept me going through this journey.

TABLE OF CONTENTS

LIST OF TABLES	10
LIST OF FIGURES	11
ABSTRACT	13
1. INTRODUCTION	16
1.1 Background and Problem Statement	16
1.2 Review of Related Studies and Knowledge Gaps	18
1.2.1 Utility compliance checking	18
1.2.2 Utility ontology development	20
1.2.3 Interpretation of utility regulations.....	21
1.3 Research Goal and Objectives	22
1.4 Research Significance and Contributions	25
1.5 Dissertation Organization	27
2. SEMANTIC APPROACH TO COMPLIANCE CHECKING OF UNDERGROUND UTILITIES	28
2.1 Introduction.....	28
2.2 Background and Review of Related Studies.....	31
2.2.1 The interoperability issue of geospatial data and the semantic solution	31
2.2.2 Automated compliance checking and the ontology-based approach	33
2.3 Proposed Semantic Approach to Compliance Checking of Underground Utilities	35
2.3.1 Ontology interlinking module.....	36
2.3.2 RDF conversion module	37
2.3.3 Compliance checking module	37
2.4 Ontology Development and Interlinking	38
2.4.1 UPO	39
2.4.2 TOO	40
2.4.3 GEO	41
2.4.4 USRO.....	42
2.4.5 Cross-ontology linkage	43
2.5 RDF Data Conversion.....	44

2.5.1	RDF convertor for geospatial data	44
2.5.2	RDF convertor for textual data.....	46
2.6	Utility Compliance Checking.....	50
2.7	Implementation Architecture	55
2.8	Case Illustration.....	56
2.9	Discussion	63
2.10	Summary and Conclusions	65
3.	TOWARDS A DOMAIN ONTOLOGY FOR UTILITY INFRASTRUCTURE: COUPLING THE SEMANTICS FROM CITYGML UTILITY NETWORK ADE AND DOMAIN GLOSSARIES	66
3.1	Introduction.....	67
3.2	Background and Related Studies.....	68
3.2.1	The interoperability and ontology in the utility infrastructure domain	68
3.2.2	Natural language processing in ontology development.....	70
3.3	Study Objectives and Contributions.....	72
3.4	Development of a Domain Ontology for Utility Infrastructure	73
3.4.1	Base ontology development	74
3.4.1.1	CityGML Utility Network ADE	74
3.4.1.2	Base ontology in OWL.....	75
3.4.2	Ontology learning.....	77
3.4.2.1	Term extraction	77
3.4.2.2	Semantic relationship classification	79
3.4.3	Ontology enrichment	84
3.4.3.1	Incorporation of key terms.....	85
3.4.3.2	Incorporation of mentioned terms.....	87
3.4.3.3	Semantic refinement.....	87
3.5	Experimentation and Case Demonstration.....	89
3.5.1	Term extraction	89
3.5.2	Semantic relationship classification	91
3.5.3	Case demonstration.....	94
3.6	Summary and Conclusions	98

4. ONTOLOGY AND RULE-BASED NATURAL LANGUAGE PROCESSING APPROACH FOR INTERPRETING TEXTUAL REGULATIONS ON UNDERGROUND UTILITY INFRASTRUCTURE	100
4.1 Introduction.....	100
4.2 Background and Review of Related Studies.....	102
4.2.1 Automation in the interpretation of regulatory documents.....	102
4.2.2 NLP-based information extraction	103
4.3 Ontology and Rule-based Approach for the Interpretation of Utility Regulations	105
4.3.1 Text preprocessing.....	106
4.3.2 Annotation of regulatory sentences	107
4.3.2.1 The annotation schema for regulatory sentences	108
4.3.2.2 Use of ontologies.....	109
4.3.2.3 Use of gazetteer lists.....	111
4.3.2.4 Use of syntactic patterns.....	111
4.3.3 Analysis of target information elements	112
4.3.3.1 Identification of target information elements and the structured representation ...	113
4.3.3.2 Analysis of syntactic relationships among target information elements	114
4.3.4 Extraction of target information elements	115
4.3.4.1 Extraction of SRTs.....	115
4.3.4.2 Extraction and assignment of attributes	118
4.3.4.3 Extraction and assignment of deontic operator and negation indicators.....	119
4.3.5 Formalization of target information elements.....	119
4.3.5.1 Semantic formalization via ontologies	120
4.3.5.2 Logic representation via deontic logic	121
4.4 Implementation.....	123
4.5 Experiments and Results.....	126
4.5.1 Experiment setup – source text selection and ontology development.....	126
4.5.2 Development of text patterns for information extraction	126
4.5.3 Evaluation, results, and analysis.....	127
4.6 Discussion	130
4.7 Summary and Conclusions	131

5. CONCLUSIONS.....	132
5.1 Summary.....	132
5.2 Limitations and Future Research.....	133
REFERENCES	137

LIST OF TABLES

Table 2.1. Mappings from spatial indicators to spatial functions	54
Table 2.2. The sentences of spatial constraints.....	58
Table 2.3. Generated SPARQL queries for compliance checking	59
Table 2.4. Comparison results	63
Table 3.1. UML-to-OWL mappings (partial) for re-structuring the <i>Network Components</i> module	76
Table 3.2. The specific semantic relationships, descriptions, and illustrative examples	81
Table 3.3. A partial list of keywords for each class under AbstractNetworkFeature	86
Table 3.4. Evaluation results for term extraction.....	90
Table 3.5. Evaluation results for semantic relationship classification	92
Table 4.1. Example patterns for SI and SRT and their corresponding matched texts	116
Table 4.2. Number of patterns for sentence annotation and information extraction	127
Table 4.3. Evaluation results for the test set.....	128

LIST OF FIGURES

Figure 1.1. Various types of utility pipes sharing the underground space	17
Figure 1.2. (a) waterline exposed by adjacent sewer collapse; (b) collateral utility damage due to sewer collapse; and (c) waterline contamination and erosion due to adjacent sewer line break...	18
Figure 1.3. Research overview	23
Figure 2.1. An example of RDFS/OWL ontology and its RDF instance.....	32
Figure 2.2. The overall framework for utility compliance checking	36
Figure 2.3. The architecture of RDF convertors.....	37
Figure 2.4. Utility product ontology	40
Figure 2.5. Transportation object ontology	41
Figure 2.6. Geometry ontology.....	42
Figure 2.7. Utility spatial rule ontology	43
Figure 2.8. Cross-ontology linkage.....	44
Figure 2.9. Conversion from Shapefile to RDF.....	45
Figure 2.10. An excerpt of the resulting RDF output in Turtle format.....	46
Figure 2.11. Examples of utility spatial constraints.....	47
Figure 2.12. Extraction of spatial cognitive-linguistic elements from spatial constraint sentences	48
Figure 2.13. Mapping process of TripleText.....	49
Figure 2.14. An excerpt of the resulting RDF output in Turtle format.....	49
Figure 2.15. An example of SPARQL query for retrieval of spatial constraint information.....	50
Figure 2.16. A partial view of the semantic resource for urban infrastructure domain	51
Figure 2.17. The SPARQL query for selecting the utility product of water line	52
Figure 2.18. Code excerpts of extended SPARQL functions using SPIN	53
Figure 2.19. An illustrative example of generated SPARQL queries	55
Figure 2.20. Implementation architecture and data flow	56
Figure 2.21. A map view of urban infrastructure in the AOI	57
Figure 2.22. A partial graph view of the conversion result	59
Figure 3.1. The development process of the utility ontology	73

Figure 3.2. Modules of the CityGML Utility Network ADE and the UML diagram (partial) of Network Components [68]	75
Figure 3.3. The base ontology (Network Components module) in graphs (partial)	77
Figure 3.4. Term extraction from the textual definitions	78
Figure 3.5. Linear-chain CRF graph structure.....	79
Figure 3.6. Feature representation for each word	79
Figure 3.7. The dependency parsing result of an example sentence	82
Figure 3.8. (a) The overall architecture for semantic relationship classification, (b) the LSTM networks for feature learning along the SDPs, and (c) the structure of an LSTM unit	83
Figure 3.9. Ontology enrichment process	85
Figure 3.10. The resulting ontology with incorporated semantics (partially)	88
Figure 3.11. Confusion matrix for term extraction in the test set	90
Figure 3.12. Confusion matrix for semantic relationship classification in the test set	93
Figure 3.13. Ontology learning from glossaries	95
Figure 3.14. Hierarchies of the classes, object properties, data properties, and datatypes in the resulting ontology	96
Figure 4.1. Proposed approach for the interpretation of utility regulations	106
Figure 4.2. NLP pipeline for text preprocessing.....	107
Figure 4.3. An annotated example of regulatory sentence	108
Figure 4.4. A partial view of SO and UPO.....	110
Figure 4.5. An example JAPE rule	112
Figure 4.6. An example of regulatory sentence represented as HSLC 7-tuples	114
Figure 4.7. The syntactic relationships among the 7-tuple elements and the extraction bases ...	115
Figure 4.8. Extraction process of HSLC SRTs.....	117
Figure 4.9. The relationships among the SO concepts and the formal spatial relations	120
Figure 4.10. Implementation architecture for information extraction	123
Figure 4.11. Illustrative examples of information extraction and formalization.....	125
Figure 5.1. Future system implementation	135
Figure 5.2. Spoken language-based human-machine interaction	136

ABSTRACT

Underground utilities must comply with the requirements stipulated in utility regulations to ensure their structural integrity and avoid interferences and disruptions of utility services. Noncompliance with the regulations could cause disastrous consequences such as pipeline explosion and pipeline contamination that can lead to hundreds of deaths and huge financial loss. However, the current practice of utility compliance checking relies on manual efforts to examine lengthy textual regulations, interpret them subjectively, and check against massive and heterogeneous utility data. It is time-consuming, costly, and error prone. There remains a critical need for an effective mechanism to help identify the regulatory non-compliances in new utility designs or existing pipelines to limit possible negative impacts. Motivated by this critical need, this research aims to create an intelligent, knowledge-based method to automate the compliance checking for underground utilities.

The overarching goal is to build semantic intelligence to enable knowledge-based, automated compliance checking of underground utilities by integrating semantic web technologies, natural language processing (NLP), and domain ontologies. Three specific objectives are: (1) designing an ontology-based framework for integrating massive and heterogeneous utility data for automated compliance checking, (2) creating a semi-automated method for utility ontology development, and (3) devising a semantic NLP approach for interpreting textual utility regulations. Objective 1 establishes the knowledge-based skeleton for utility compliance checking. Objectives 2 and 3 build semantic intelligence into the framework resulted from Objective 1 for improved performance in utility compliance checking.

Utility compliance checking is the action that examines geospatial data of utilities and their surroundings against textual utility regulations. The integration of heterogeneous geospatial data of utilities as well as textual data remains a big challenge. Objective 1 is dedicated to addressing this challenge. An ontology-based framework has been designed to integrate heterogeneous data and automate compliance checking through semantic, logic, and spatial reasoning. The framework consists of three key components: (1) four interlinked ontologies that provide the semantic schema to represent heterogeneous data, (2) two data convertors to transform data from proprietary formats into a common and interoperable format, and (3) a reasoning mechanism with spatial extensions

for detecting non-compliances. The ontology-based framework was tested on a sample utility database, and the results proved its effectiveness.

Two supplementary methods were devised to build the semantic intelligence in the ontology-based framework. The first one is a novel method that integrates the top-down strategy and NLP to address two semantic limitations in existing ontologies for utilities: lack of compatibility with existing utility modeling initiatives and relatively small vocabulary sizes. Specifically, a base ontology is first developed by abstracting the modeling information in CityGML Utility Network ADE through a series of semantic mappings. Then, a novel integrated NLP approach is devised to automatically learn the semantics from domain glossaries. Finally, the semantics learned from the glossaries are incorporated into the base ontology to result in a domain ontology for utility infrastructure. For case demonstration, a glossary of water terms was learned to enrich the base ontology (formalized from the ADE) and the resulting ontology was evaluated to be an accurate, sufficient, and shared conceptualization of the domain.

The second one is an ontology- and rule-based NLP approach for automated interpretation of textual regulations on utilities. The approach integrates ontologies to capture both domain and spatial semantics from utility regulations that contain a variety of technical jargons/terms and spatial constraints regarding the location and clearance of utility infrastructure. The semantics are then encoded into pattern-matching rules for extracting the requirements from the regulations. An ontology- and deontic logic-based mechanism have also been integrated to facilitate the semantic and logic-based formalization of utility-specific regulatory knowledge. The proposed approach was tested in interpreting the spatial configuration-related requirements in utility accommodation policies, and results proved it to be an effective means for interpreting utility regulations to ensure the compliance of underground utilities.

The main outcome of this research is a novel knowledge-based computational platform with semantic intelligence for regulatory compliance checking of underground utilities, which is also the primary contribution of this research. The knowledge-based computational platform provides a declarative way rather than the otherwise procedural/hard-coding implementation approach to automate the overall process of utility compliance checking, which is expected to replace the conventional costly and time-consuming skill-based practice. Utilizing this computational platform for utility compliance checking will help eliminate non-compliant utility designs at the

very early stage and identify non-compliances in existing utility records for timely correction, thus leading to enhanced safety and sustainability of the massive utility infrastructure in the U.S.

1. INTRODUCTION

Underground utilities must comply with the requirements stipulated in utility regulations to ensure their structural integrity and avoid interferences and disruptions of utility services. Noncompliance with the regulations could lead to utility incidents such as pipeline explosion and pipeline contamination, with disastrous consequences of property damages, environmental pollution, and personnel injuries and fatalities. Utility compliance checking is the action that examines the geospatial data of utilities and their surroundings against utility regulation data to identify the regulatory non-compliances in utility designs or existing records to limit possible negative impacts. However, the current practice of utility compliance checking relies on manual efforts to examine lengthy textual regulations, interpret them subjectively, and check against massive and heterogeneous utility data. It is time-consuming, costly, and error prone. This research aims to create an intelligent, knowledge-based method to automate the compliance checking of underground utilities. This chapter provides an overview of this research.

1.1 Background and Problem Statement

Underground utilities provide the core services such as water, electricity, gas, and fiber networks to the society. Their physical networks - drinking water transmission and distribution, wastewater collection and stormwater drainage systems, natural gas, telecommunications, television and electrical power – all share the underground space, as shown in Figure 1.1.

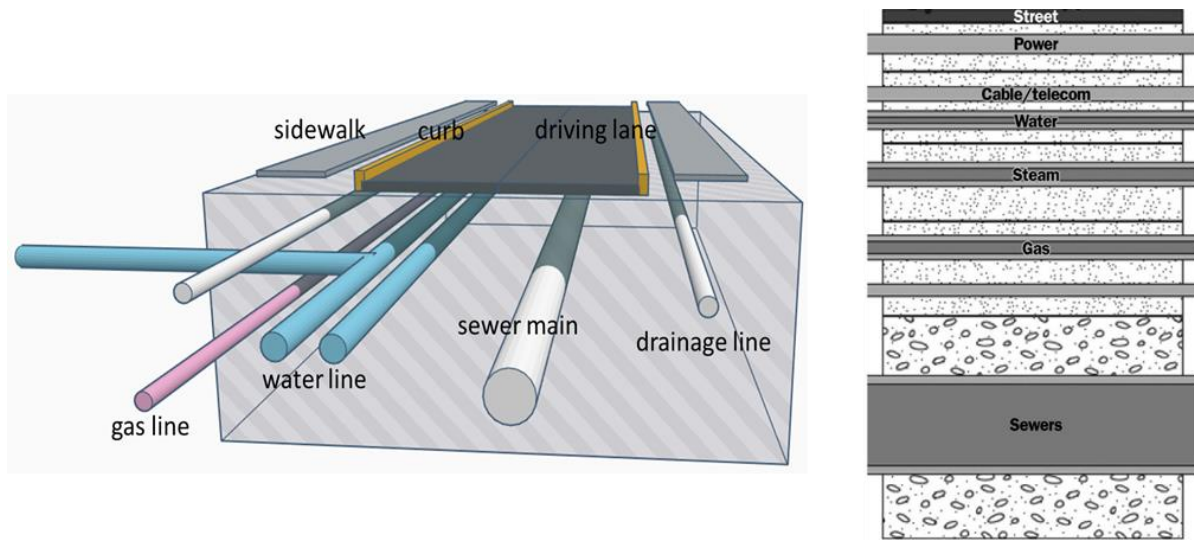


Figure 1.1. Various types of utility pipes sharing the underground space

Utility regulatory documents such as design guidelines, codes, and manuals of practice stipulate the spatial constraints among utilities and their surroundings (e.g., road networks and urban developments) to ensure their structural integrity and avoid interferences and disruptions of utility services. For example, a minimum depth of cover of utility pipes under the roadway is specified to help maintain the structural integrity of the pipeline throughout its service life. Another example is the adequate separation between pipelines to reduce the potential of pipeline failure caused by a leak or failure of its neighboring pipeline. Figure 1.2 shows cases of adjacent pipeline failure because of inadequate separation. Noncompliance with these spatial constraints could lead to utility incidents such as pipeline explosion and pipeline contamination, with disastrous consequences of property damages, environmental pollution, and personnel injuries and fatalities [1,2]. For instance, the noncompliance with the regulated minimum separation between the oil pipeline and an urban storm drain resulted in accelerated pipeline corrosion, leakage and the following explosion in the City of Qingdao, China in November 2013, which caused 62 fatalities, 136 injures, and 2,000 tons of oil leakage into the sea [1]. The direct economic loss amounted to US\$122.23 million. Similar deficiencies were found in the 2008 Rancho Cordova pipeline explosion and the 2010 San Bruno pipeline explosion [3,4].

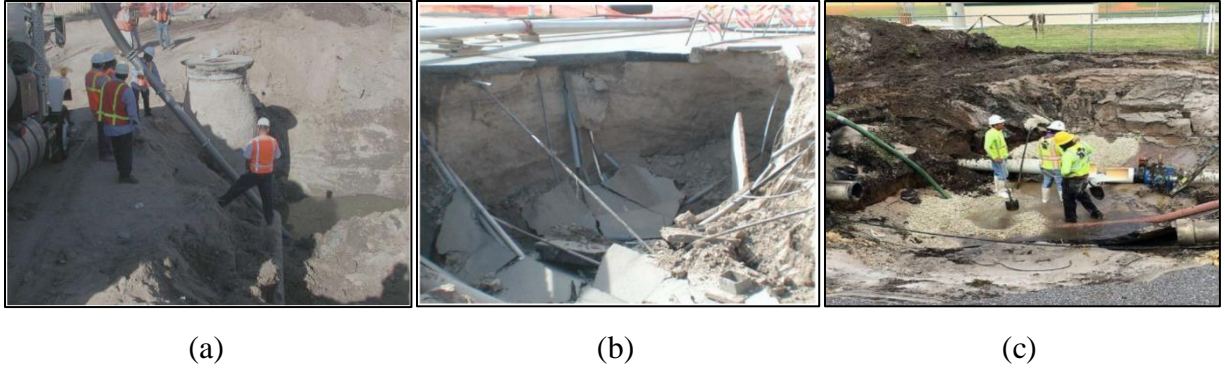


Figure 1.2. (a) waterline exposed by adjacent sewer collapse; (b) collateral utility damage due to sewer collapse; and (c) waterline contamination and erosion due to adjacent sewer line break

The recurrent utility incidents emphasize the importance of spatial compliance with utility regulations. However, the current practice of utility compliance checking relies on manual efforts to examine lengthy textual regulations, interpret them subjectively, and check against massive and heterogeneous utility data. It is time-consuming, costly, and error prone. There remains a critical need for a compliance checking mechanism to help identify spatial non-compliances in utility new designs or existing records for timely correction to limit possible negative impacts. Not meeting this need represents an important problem because, without compliance, inadequately designed utilities will continue to be built and existing, deficient utilities are unlikely to be retrofitted appropriately.

1.2 Review of Related Studies and Knowledge Gaps

This section reviews the related studies and highlights the knowledge gaps. The related studies can be divided into three areas of knowledge each of which is respectively discussed in the following sub-sections. Limitations and what are needed to overcome the limitations are also specified.

1.2.1 Utility compliance checking

Utility compliance checking is the action that examines geospatial data of utilities and their surroundings against utility regulations [1]. However, most of the geospatial data remain in various geographic formats (e.g., ESRI Shapefiles [5]) or propriety databases (e.g., Oracle Spatial [6] or

PostGIS [7]) while utility regulations are typically textual documents. Due to the lack of unified standards in the utility domain, data sharing and exchange between different information systems become very challenging. A mechanism that integrates heterogeneous geospatial data as well as regulation data is the critical prerequisite to utility compliance checking.

Research efforts have been conducted to develop open data standards to handle the mismatch between heterogeneous data formats. For instance, Industry Foundation Class (IFC) is the open standard format for BIM by establishing interoperability in the construction industry [8] while CityGML is the standard data model established by Open Geospatial Consortium (OGC) for exchange of geospatial data and the interoperability between 3D GIS systems [9]. However, these open standards are limited to the level of syntax and structure. Although a rich set of concepts/classes and relations are provided in the open standards, detailed, accurate, consistent, sound, and meaningful distinctions are not made among the concepts/classes and relations [10,11]. The lack of such declarative semantics imposes big challenges on data exchange between disparate sources that use different sets of vocabularies [2,10]. Therefore, there is a need to design an effective data exchange mechanism that can facilitate semantic integration of heterogeneous utility data for the purpose of compliance checking.

In terms of the implementation approach of compliance checking, computer-based compliance checking in the Architecture, Engineering, and Construction (AEC) domain traces back to 1960s when Fenves et al. [12] proposed a decision table approach to aid engineering design for conformance with American Institute of Steel Construction (AISC) specifications. Over the past decades, there have been significant advancements to automate the compliance checking process, such as the checking of building envelope performance [13], fire code compliance [14–17], building safety design [18], building evacuation [19], building structural design [20], and construction inspection and quality control [21]. Computational implementation and tools have also been developed by practitioners and software developers, e.g., DesignCheck, Solibri Model Checker, ePlanCheck, and SMARTCodes [22]. However, most compliance checking environments seem forced to rely on a hard-coding implementation approach, which involves much arbitrary programming work and is unreachable for anyone but system programmers, whereas a declarative implementation approach that is based on a rule language is argued as the better choice for compliance checking environments [11,22–24]. Therefore, there is another need to

design a more transparent mechanism for utility compliance checking that are easy-to-understand and simple-to-implement even by non-experts.

Recently, ontology has emerged as a promising tool to achieve semantic interoperability over fragmented, heterogeneous application environments [11]. An ontology describes the concepts, relationships, data properties and restrictions within a domain in a machine-readable manner [25,26], which can be utilized as the shared data format for each source to integrate data in heterogeneous formats. An increasing number of information management/exchange applications in construction have been relying on ontologies to support data interoperability, flexible data exchange, distributed data management, and the development of reusable tools [11,27,28]. In the GIS community, ontology has also been exploited to integrate a large amount of heterogeneous geospatial data [29–31]. On the other hand, attributed to its logic foundation, ontology is also used in automated reasoning for compliance checking [32] [23] [33], which provides a more transparent paradigm rather than the otherwise procedural/hard-coding implementation approach. Therefore, ontology is used in this study to address the above two needs.

1.2.2 Utility ontology development

A few domain ontologies have been introduced for the utility domain [2,26,34–36]. However, they are very limited to facilitate data exchange in heterogeneous environments for the following two reasons. First, they are mainly implemented as a means for knowledge representation and neglect the compatibility with existing utility modeling initiatives [10]. Much laborious work is required to align the semantic schemas in the ontologies with the data schemas in various utility models for data exchange [26,35]. Second, their semantic vocabularies of domain terms and semantic relationships are relatively too small to interpret the meaning of data and avoid mismatches/no matches when integrating a multitude of data that have different terms [37]. There is a critical need in the utility domain for an ontology that can be utilized as the shared and reliable knowledge model to facilitate a high degree of interoperability.

Several ontology development methodologies have been suggested [38,39]. They all include five key steps: (1) purpose and scope definition, (2) taxonomy building, (3) relation modeling, (4) ontology coding, and (5) ontology evaluation. The five-step method for ontology development requires significant manual efforts on knowledge retrieval and ontology construction and validation. In attempts to reduce laborious work on ontology development, researchers have

sought to design natural language processing (NLP) algorithms to build ontologies from a corpus of natural language text. NLP deploys artificial intelligence to enable computers to understand, create, and analyze human languages [40]. It contributes to ontology development in automated extraction of ontology contents – concepts and relations from textual documents. Since it is challenging to directly build an ontology from the extracted concepts and relations (higher textual analysis and more human work are required) [41], most studies end up building plain (or unstructured) dictionaries that simply archive the extracted ontology contents [37,42]. A few studies have adopted a top-down strategy to build ontology from the extracted concepts and relations [43,44]. Existing semantic models (taxonomies/ontologies) are first selected as bases, and enrichment follows by using the contents extracted from textual documents. For instance, Zhang and EI-Gohary [43] utilized rule-based NLP to extract concepts/relationships from regulatory documents and extended the existing IFC taxonomy with the extracted contents.

The top-down strategy can save significant time and effort in building the knowledge skeletons of the ontology – the ontology directly inherited the semantics (formal definitions of classes and relations) provided by the existing semantic models. As such, this study adopts the top-down strategy and devises a novel integrated NLP approach (used to extract the semantics from textual documents for ontology enrichment) to develop an ontology for the utility infrastructure domain.

1.2.3 Interpretation of utility regulations

Utility regulations stipulate the spatial configurations among underground utility networks and their surroundings to avoid interferences and disruptions of utility services [1,2,34]. In the current practice, practitioners perform compliance checking, with the aim of detecting violations in designs and existing records, by manually going through the lengthy textual regulations, interpreting them subjectively based on their knowledge and experience, and checking massive and heterogeneous utility data against them [1,45]. This practice is neither efficient, nor sustainable, attributed to the large size of and the heterogeneity in utility regulatory documents [1] and the heavy reliance of the interpretation on human knowledge and subjective judgement – different interpreters might entail different meanings from the same clause [15]. Therefore, there is a critical need for an automated approach for the consistent interpretation of textual regulations on underground utilities to ensure the compliance of underground utility infrastructure.

A number of approaches have been attempted to automate the interpretation process for regulatory documents in the Architecture, Engineering, and Construction (AEC) domain. Examples include the use of hypertext and hypermedia to aid in navigating regulatory documents [46,47] and the use of document markup techniques to assist in analyzing the semantic structure of target regulatory requirements [48]. Nevertheless, these methods require intense manual efforts on annotating regulatory documents for further interpretation [24,48]. Natural Language Processing (NLP) methods have emerged in recent years to automate the extraction of requirements from textual documents such as building codes [49,50] and utility regulations [1]. Further, NLP has also been attempted to transform the extracted requirements into a structured format (i.e., logic clauses) for compliance checking [51]. Technical challenges in automating the interpretation of utility regulations include 1) heterogeneous technical terminologies – utility regulations contain a variety of technical terms since different disciplines and communities of practice may adopt different sets of vocabularies to describe their utility assets, and 2) the dominance of spatial constraints in utility regulations regarding location and clearance for the purposes of infrastructure safety, maintainability, and constructability, and public health and safety [2,34]. Consequently, a successful NLP method for the efficient and consistent interpretation of utility regulations must have the capacity to address the heterogeneity of technical terminologies and understand the spatial semantics from natural language.

Recently, ontologies have been integrated into NLP to capture the semantics from texts [52–55]. It is reported that the use of ontology yields higher performance in information extraction for a specific domain [49,50,55]. This study also integrates ontologies into NLP to help capture both domain and spatial semantics in utility regulations and further interprets them as logic clauses for supporting utility compliance checking.

1.3 Research Goal and Objectives

The overarching goal is to build semantic intelligence to enable knowledge-based, automated compliance checking of underground utilities by integrating semantic web technologies, natural language processing (NLP), and domain ontologies. To achieve this goal, three specific objectives are formulated: (1) designing an ontology-based framework for integrating massive and heterogeneous utility data for automated compliance checking, (2) creating a semi-automated

method for utility ontology development, and (3) devising a semantic NLP approach for interpreting textual utility regulations. Objective 1 establishes the knowledge-based skeleton for utility compliance checking. Objectives 2 and 3 build semantic intelligence into the framework resulted from Objective 1 for improved performance in utility compliance checking. Figure 1.3 presents the overview of the research.

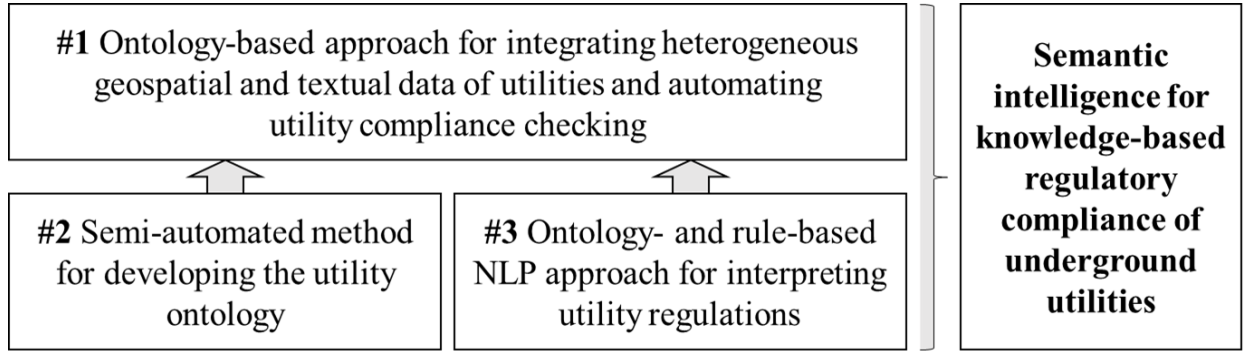


Figure 1.3. Research overview

The first objective is to develop an ontology-based framework for integrating heterogeneous geospatial and textual data of utilities and enabling automated compliance checking of underground utilities through semantic, logic, and spatial reasoning. The framework will be composed of the following three key components: (1) the ontology interlinking module that consists of four interlinked ontologies to provide the semantic schema for the representation of heterogeneous data relevant to utility compliance checking, (2) the RDF conversion module that contains two data convertors for the conversion of heterogeneous data from proprietary formats into a common and interoperable format following the semantic schema, and (3) the compliance checking module - a query mechanism with spatial extensions for the detection of utility noncompliance. The framework has two technical challenges. First, the development of the four ontologies requires significant time and effort to define their concepts, relationships, and knowledge skeletons, and it is never possible to rely on human effort to develop an ontology that has a sufficient size of semantic vocabulary. Second, the processing of utility spatial constraints requires manual annotation to prepare the checking rules for the framework, and it is always imperative to develop automated methods for rule preparation from pure texts to support fully automated compliance checking. As such, Objectives 2 and 3 are dedicated to address these two challenges, respectively. The work in Objective 2 automates the development of the utility product

ontology (UPO) – an essential ontology in the ontology interlinking module, and the work in Objective 3 automates the extraction and formalization of spatial rules from textual utility regulatory documents – critical components in the RDF data conversion and compliance checking module. Jointly, they build the semantic intelligence for the framework in Objective 1 for an improved efficacy in utility compliance checking.

The second objective is to devise a novel method to develop a utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain. The novel method will integrate a top-down strategy and NLP to develop the desired ontology from CityGML Utility Network ADE (a candidate open standard for modeling utility networks) and domain glossaries (lists of utility-specific terms and their textual definitions). The third objective is to design an ontology- and rule-based NLP approach to automate the interpretation of utility regulations – extracting the requirements from the regulations and further formalizing them into logic clauses – for supporting automated compliance checking of underground utilities. The approach will rely on ontologies to capture both domain and spatial semantics in utility regulations and encode pattern-matching rules for information extraction. A mechanism will also be designed by integrating ontologies and deontic logic (DL) to facilitate the semantic and logic-based formalization of utility-specific regulatory knowledge.

To summarize, Objective 1 provides an ontology-based framework for utility compliance checking, under which the overall process is implemented in a more transparent manner that is easy-to-understand and simple-to-develop even by non-experts. It is likely to shift the skill-based activity to a knowledge-based paradigm. Achieving Objective 2 results in a better option of interoperability facilitator – the utility ontology, which is an essential ontology in the ontology interlinking module of Objective 1 to facilitate semantic integration of heterogeneous utility data for an improved efficacy in compliance checking. Objective 2 also offers an automated method for ontology enrichment (in terms of the semantic vocabulary) from domain glossaries to keep us with new semantics. Such an ontology with an expandable semantic vocabulary will enable the semantic interpretation of data even when integrating a multitude of data from different sources that use different vocabulary sets. Achieving Objective 3 results in an automated, end-to-end NLP pipeline for interpreting the lengthy utility regulations, which can automate the extraction and formalization of spatial rules from textual utility regulatory documents in the text-to-RDF

conversion and compliance checking modules of Objective 1. A system that consists of these three objectives is expected to achieve higher levels of semantic intelligence, automation, and efficiency in utility compliance checking, which will be demonstrated in future.

1.4 Research Significance and Contributions

Underground utilities must comply with the requirements stipulated in utility regulations to ensure their structural integrity and avoid interferences and disruptions of utility services [1,2]. Noncompliance with the regulations could cause disastrous consequences such as pipeline explosion and pipeline contamination, that leads to hundreds of death and huge financial loss [1]. There remains a critical need for an effective mechanism to help identify the regulatory non-compliances in utility designs or existing records to limit possible negative impacts. The National Academy of Engineering identified “restore and improve urban infrastructure” as a grand challenge for Engineering in the 21st century. Not meeting this need represents an important problem because, without compliance, inadequately designed utilities will continue to be built and existing, deficient utilities are unlikely to be retrofitted appropriately. The research is expected to create a knowledge-based, computational method with semantic intelligence to automate the compliance checking of underground utilities. Deploying this computational method will help eliminate non-compliant utility designs at the very early stage and identify non-compliances in existing utility records for timely correction, thus leading to enhanced safety and sustainability of the massive utility infrastructure in the U.S.

The primary contribution of this research is the knowledge-based computational platform with semantic intelligence for regulatory compliance checking of underground utilities. The knowledge-based computational platform provides a declarative way rather than the otherwise procedural/hard-coding implementation approach to automate the overall process of utility compliance checking, which is expected to replace the conventional costly and time-consuming skill-based utility compliance checking practice. Specifically, this research contributes to the body of knowledge in the following areas.

First, the research develops a novel, ontology-based semantic approach that can facilitate the semantic integration of heterogeneous data and enable the automated compliance checking of underground utilities through semantic, logic, and spatial reasoning. The approach relies on four interlinked ontologies and two data convertors to address the issue of data heterogeneity in the

utility infrastructure domain. The approach also advances existing ontology-based compliance checking efforts by adding more advanced reasoning capabilities (e.g., spatial reasoning), which can support a wider range of application scenarios. Moreover, the approach enables a more transparent implementation of utility compliance checking that are easy-to-understand and simple-to-implement even by non-experts, which is likely to shift this skill-based activity to a knowledge-based paradigm.

The next contribution is a novel integrated method for automatically building the utility ontology from CityGML Utility Network ADE and domain glossaries. The method that integrates a top-down strategy and NLP can significantly reduce the laborious work during the process of ontology development. The method can also be adapted to ontology development for other domains. The integrated NLP enables fully automated extraction of ontology contents from domain glossaries, which can help maintain the ontology to keep up with the growth of new domain knowledge. Besides, the developed ontology is a superior interoperability facilitator for the utility infrastructure domain as compared to the existing ones. Specifically, the ontology is semantically compatible with the modeling practice in the utility industry; and also, the ontology has an enriched semantic vocabulary (which can be expanded from domain glossaries in timely and automated manners), which can facilitate the semantic integration of data between disparate sources that use different sets of vocabularies. Relying on this ontology for utility compliance checking, semantic intelligence can be enabled, thus leading to an improved efficiency during compliance checking.

Third, the research develops an NLP approach for interpreting the textual regulations on underground utility. The NLP approach integrates ontologies to allow for the extraction and formal representation of domain-specific and spatial information from utility regulations. It has the capacity to capture both domain and spatial semantics from natural language. Further, a mechanism is designed to transform the extracted/formalized information (unstructured information pieces) into logic clauses, thus providing an end-to-end pipeline for interpreting utility regulations. The approach enables the automated interpretation of utility regulations to provide ready-to-use logic rules for utility compliance checking, thus improving the level of automation in utility compliance checking.

1.5 Dissertation Organization

This dissertation is organized into five chapters and follows the “multiple publications” formats. Each of the Chapters 2, 3, and 4 has its own introduction, literature review, methodology, implementation and results, and conclusion sections. Significant portions of these chapters have been published or submitted for review and publication in peer reviewed journals. Chapter 1 introduces the background, highlights the problem statement and limitations in related studies, and discusses the research objectives, significance and contributions.

Chapter 2 presents the development of the ontology-based framework for integrating heterogeneous geospatial and textual data of utilities and enabling automated compliance checking of underground utilities through semantic, logic, and spatial reasoning. *This work was previously published in Automation in Construction. This chapter is re-printed with permission from Vol 109, Xin Xu and Hubo Cai, “Semantic approach to compliance checking of underground utilities”, 03006, Copyright Elsevier (2019). Table titles and figure captions have been modified to maintain the form of the dissertation.*

Chapter 3 describes a novel method to develop a utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain. *This work is under review in ASCE Journal of Computing in Civil Engineering, 2020, Xin Xu and Hubo Cai. “Towards a domain ontology for utility infrastructure: coupling the semantics from CityGML Utility Network ADE and domain glossaries”. Table titles and figure captions have been modified to maintain the form of the dissertation.*

Chapter 4 discusses the design of an NLP approach to automate the interpretation of utility regulations – extracting the requirements from the regulations and further formalizing them into logic clauses – for supporting automated compliance checking of underground utilities. *This work is under review in Advanced Engineering Informatics, 2020, Xin Xu and Hubo Cai. “Ontology and Rule-based Natural Language Processing Approach for Interpreting Textual Regulations on Underground Utility Infrastructure”. Table titles and figure captions have been modified to maintain the form of the dissertation.*

The final chapter, Chapter 5, concludes the dissertation with the major findings and future research opportunities.

2. SEMANTIC APPROACH TO COMPLIANCE CHECKING OF UNDERGROUND UTILITIES

This chapter presents the development of the ontology-based framework for integrating heterogeneous geospatial and textual data of utilities and enabling automated compliance checking of underground utilities through semantic, logic, and spatial reasoning. The framework consists of the following key components: (1) four interlinked ontologies that provide the semantic schema for the representation of heterogeneous data relevant to utility compliance checking, (2) two data convertors for the conversion of heterogeneous data from proprietary formats into a common and interoperable format, and (3) a reasoning mechanism with spatial extensions for the detection of utility noncompliance. The ontology-based framework was tested on a sample utility database, and the results demonstrate the success of the framework in the integration of heterogeneous utility data from multiple sources and automated detection of regulatory non-compliances in underground utilities.

This work was previously published in Automation in Construction. This chapter is re-printed with permission from Vol 109, Xin Xu and Hubo Cai, “*Semantic approach to compliance checking of underground utilities*”, 103006, Copyright Elsevier (2019). Table titles and figure captions have been modified to maintain the form of the dissertation.

2.1 Introduction

Utility regulatory documents such as design guidelines, codes, and manuals of practice stipulate the spatial constraints among utilities and their surroundings (e.g., road networks and urban developments) to ensure their structural integrity and avoid interferences and disruptions of utility services. For example, a minimum depth of cover of pipelines under the roadway is specified to help maintain the structural integrity of the pipeline throughout its service life. Another example is the stated location-preference for utility facilities such as manholes, vaults, and pits to facilitate service access and minimize disruptions to transportation facilities. Noncompliance with these spatial constraints could lead to utility incidents such as pipeline explosion and pipeline contamination, with disastrous consequences of property damages, environmental pollution, and personnel injuries and fatalities [1,2]. For instance, the noncompliance with the regulated

minimum separation between the oil pipeline and an urban storm drain resulted in accelerated pipeline corrosion, leakage and the following explosion in the City of Qingdao, China in November 2013, which caused 62 fatalities, 136 injures, and 2,000 tons of oil leakage into the sea [56]. The direct economic loss amounted to US\$122.23 million. Similar deficiencies were found in the 2008 Rancho Cordova pipeline explosion and the 2010 San Bruno pipeline explosion [3]. The recurrent utility incidents emphasize the importance of spatial compliance with utility regulations. There remains a critical need for a compliance checking mechanism to help identify spatial non-compliances in utility new designs or existing records for timely correction to limit possible negative impacts. Not meeting this need represents an important problem because, without compliance, inadequately designed utilities will continue to be built and existing, deficient utilities are unlikely to be retrofitted appropriately.

Utility compliance checking is the action that examines geospatial data of utilities and their surroundings against utility regulations [1]. However, most of the geospatial data remain in various geographic formats (e.g., ESRI Shapefiles [5]) or DBMSs (e.g., Oracle Spatial [6] or PostGIS [7]) while utility regulations are typically textual documents. Due to a lack of unified standards in the utility domain, data sharing and exchange between different information systems become very challenging. A mechanism that integrates heterogeneous geospatial data as well as textual data is the critical prerequisite to the compliance checking of underground utilities.

Research efforts have been conducted to develop open data standards to handle the mismatch between heterogeneous data formats. For instance, IFC is the open standard format for BIM by establishing interoperability in the construction industry [8] while CityGML is the standard data model established by Open Geospatial Consortium (OGC) for exchange of geospatial data and the interoperability between 3D GIS systems [9]. However, these open standards are limited to the level of syntax and structure. Although a rich set of concepts/classes and relations are provided in the open standards, detailed, accurate, consistent, sound, and meaningful distinctions are not made among the concepts/classes and relations [10,11]. The lack of such declarative semantics imposes big challenges on data exchange between disparate sources that use different sets of vocabularies [2,10]. Recently, ontology has emerged as a promising tool to achieve semantic interoperability over fragmented, heterogeneous application environments [11]. An ontology is an explicit formalization of a shared conceptualization: “conceptualization” refers to an abstract model of the relevant concepts and relationships; “explicit” means that the types of concepts used and the

constraints on their use are explicitly defined; “formalization” refers to the fact that the ontology should be machine processable [57]. In the context of semantic web, ontology plays a key role in providing the semantic vocabulary used to annotate websites in a way meaningful for machine interpretation [58]. In a similar way, ontology can fill the semantic gap in existing open data standards by providing a shared semantic vocabulary. From the perspective of semantic web applications, ontologies are usually expressed based on logic theory using modeling languages of Resource Description Framework Schema (RDFS) [59] and Web Ontology Language (OWL) [60] developed by the World Wide Web Consortium (W3C), so that declarative semantics can be incorporated into the concepts and relationships; semantic tools (e.g., Resource Description Framework (RDF) [61], SPARQL [62]) support automated reasoning using the ontologies, and thus provide advanced services to intelligent applications. RDF, a standard data model, offers a unified format for describing individual ontology instances. It can facilitate the semantic integration of disparate and heterogeneous data. SPARQL, a query language, enables the logic-based manipulation of RDF data and when extended, supports more advanced reasoning (such as spatial reasoning) by defining custom rules [63,64]. Given that utility compliance checking requires both data integration from multiple sources and spatial reasoning, domain ontologies need to be developed to facilitate the semantic integration of heterogeneous utility data and SPARQL spatial extensions need to be added to realize the checking of spatial utility data against spatial rules.

Towards that end, this paper creates an automated compliance checking mechanism for the utility domain by combining ontology and SPARQL spatial extensions. Specifically, the following key components are involved in the proposed mechanism: 1) four interlinked ontologies that provide the semantic schema for heterogeneous data relevant to utility compliance checking, 2) two data convertors for the conversion of heterogeneous data from proprietary formats into the common and interoperable format of RDF following the semantic schema, and 3) a query mechanism with SPARQL spatial extensions for the detection of non-compliant utility instances. With this new approach, the compliance checking process is comprised of retrieval of spatial constraints, generation of SPARQL queries, execution of SPARQL queries, and reporting. An experiment on a sample utility project was conducted to determine the feasibility and effectiveness of the proposed approach in detecting utility spatial defects. Such a mechanism would help prevent utility design problems from the earliest phase and increase efficiency in managing utility existing

defects in a timely manner, thus leading to enhanced safety and sustainability of the massive utility infrastructure in the society.

2.2 Background and Review of Related Studies

This section reviews related studies with a focus on semantic approaches to addressing the interoperability issue in geospatial utility data and automating compliance checking.

2.2.1 The interoperability issue of geospatial data and the semantic solution

Due to the segmentation of the utility industry, most of the existing geospatial utility data are stored and managed in propriety databases and GIS platforms, using a variety of data models and formats [2,34]. Example utility network models include INSPIRE Utility Networks, ArcGIS Utility Networks, IFC Utility model, SEDRIS, and Pipeline ML. Data exchange and interoperability between different data formats and utility models have been a big issue [2,34].

In the GIS domain, many open standards have been developed by OGC to standardize and hence facilitate the exchange of geospatial data across different GIS applications and systems. Among the existing standardbreds, Geographical Markup Language (GML) [65] and CityGML [66] are the ones that are mostly relevant to modeling utility networks and their surroundings. GML is a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. CityGML, a GML application schema, provides an open data model for the storage and exchange of virtual 3D city models. It defines classes and relations for the most relevant topographic objects with respect to their geometrical, topological, semantical, and appearance properties [67]. CityGML can be extended to suit various infrastructure domains; CityGML Utility Network Application Domain Extension (ADE) extends the CityGML model to define the required concepts and classes for the integration of multi-utility networks into the 3D urban space. It covers the topology, topography, and functional and semantic classification of network objects [68]. While CityGML Utility Network ADE offers a potential solution to integrate various utility models (e.g., data convertors developed for converting various utility models into CityGML Utility Network ADE such as the IFC Utility model-to-CityGML Utility Network ADE convertor [69]), such an integration remains at the syntactic level – heterogenous data are structured as loosely coupled documents that are not semantically compatible since different sets

of vocabulary are used by different practitioners for describing their assets. For example, different terms such as water pipe, water conduit, and water line could have been used for a water pipe asset. While their descriptions follow the same syntax, effective communications are still missing due to the terminological inconsistency [11,29,30].

Ontology and semantic tools have proven advantages in achieving semantic interoperability [10,11,70]. An ontology is an explicit formalization of a conceptualization that provides an abstract schema consisting of formal definitions of concepts and their relationships [57]. RDFS and OWL work together to provide most basic and more expressive elements to define concepts and relationships in machine-readable and explicit format. Figure 2.1 illustrates an example, where *owl:Class* and *owl:ObjectProperty* are used to define the ontology concept of *UtilityProduct* and the relationship of *belongsToSector* respectively while *rdfs:subClassOf* is used to define the concept hierarchy. Using the Subject – Predicate – Object structure in RDF, the ontology instance of a specific utility product X belongs to a water sector is captured as *UtilityProductX* – *belongsToSector* – *WaterSector*. The triple structure is linkable to other knowledge resources represented in RDF to construct semantic networks of interconnected knowledge resources. RDFS and OWL provide a formal way to describe the semantics of classes and properties in ontology and thus ensure the semantic consistency over the semantic network.

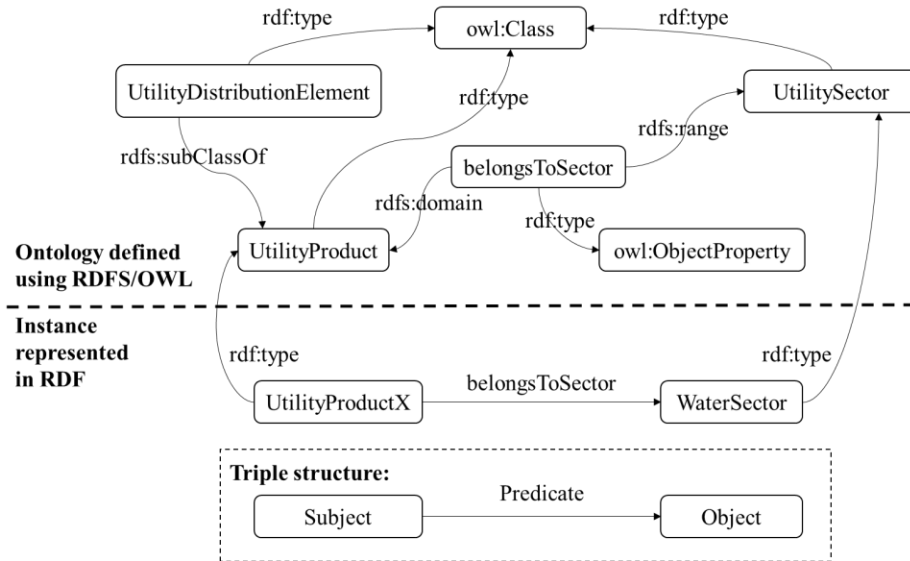


Figure 2.1. An example of RDFS/OWL ontology and its RDF instance

Attributed to their capability in semantic interoperability, ontology and semantic tools have been exploited in the GIS community to integrate a large amount of heterogeneous geospatial data. The efforts by this community have led to the CityGML ontology [29–31]. Using CityGML ontology as the central platform, Métral et al. [29,30] integrated urban infrastructure data, transportation data, and urban planning data for creating a semantically enriched 3D city model. Howell et al. [35] proposed a water knowledge management platform that incorporates semantics into the Internet of Things (IoT) to address the heterogeneity issue of web resources and support smart water networks. Howell et al. [26] further integrated building and urban semantics by developing an ontology for the domain to empower smart water solutions. OGC developed the GeoSPARQL standard to define a vocabulary for representing geospatial data in RDF [71]. The use of ontologies also benefits the GIS-BIM integration. For instance, El-Mekawy and Östman [72] relied on an intermediate reference ontology, the Unified Building Model (UBM) to achieve the bi-directional mapping between IFC and CityGML. Hor et al. [73] integrated GIS and BIM using semantic web technologies and RDF graphs for building multi-scale 3D urban models.

However, the lack of declarative semantics remains a big hurdle especially in developing ontology and semantic approaches to address the interoperability issue in the utility domain. Open utility standards such as CityGML Utility Network ADE typically do not have the semantics behind them explicitly defined and clearly communicated. The few domain ontologies that have been proposed for urban utilities [2,34] are mainly implemented as a means for knowledge management without detailing specific data elements. 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 to incorporate semantics to existing open standards through ontologies and subsequently build an ontology-based data exchange mechanism to seamlessly integrate heterogeneous utility data.

2.2.2 Automated compliance checking and the ontology-based approach

Computer-based compliance checking in the AEC domain traces back to 1960s when Fenves et al. [12] proposed a decision table approach to aid engineering design for conformance with American Institute of Steel Construction (AISC) specifications. Over the past decades, there have been significant advancements to automate the compliance checking process, such as the checking of building envelope performance [13], fire code compliance [14–17], building safety design [18],

building evacuation [19], building structural design [20], and construction inspection and quality control [21]. Computational implementation and tools have also been developed by practitioners and software developers, e.g., DesignCheck, Solibri Model Checker, ePlanCheck, and SMARTCodes [22]. Recently, Solihin et al. [74] have proposed an approach using multiple representations to achieve high-performance spatial queries on 3D BIM data, which provides the opportunity to break the conundrum of an automated rule checking system which so far has been limited to relatively simple and nonspatial rules. However, most compliance checking environments seem forced to rely on a hard-coding implementation approach, which involves much arbitrary programming work and is unreachable for anyone but system programmers, whereas a declarative implementation approach that is based on a rule language is argued as the better choice for compliance checking environments [11,22–24].

In view of that, a number of researchers explored the development of the rule language-based formalization of regulatory requirements to make the checking environments even more flexible, transparent, and portable. Domain-specific rule languages have been proposed for the construction domain such as the Building Environment Rule and Analysis (BERA) language [75] and the Drools Rule Language (DRL) [76,77] for supporting language-based compliance checking. Dimyadi and Amor [78] and Dimyadi et al. [79] also explored the potential of adapting the legal mark-up languages such as LegalDocML and LegalRuleML to accommodate the compliance checking-related requirements of the AEC domain. To achieve full automation in compliance checking, Natural Language Processing (NLP)-based approaches have also been proposed to facilitate the logic-based representation of regulatory and design information for building code checking [49,51,80]. Another more recent trend is the ontology-based approach using SPARQL queries or dedicated rule languages such as Semantic Web Rule Language (SWRL) [81], the Rule Interchange Format (RIF) [82], and N3Logic [83] to represent the requirements for semantic and logic-based reasoning. For example, Yurchyshyna and Zarli [32] proposed an ontology-based approach for the conformance checking of construction projects, in which the regulatory requirements were represented in the form of SPARQL queries. Pauwels et al. [23] used N3Logic to create the rules from acoustic performance regulations to support the semantic rule checking in BIM. Zhong et al. [33] proposed an ontology to automate the construction quality inspection and evaluation, where regulation constraints are modeled as OWL axioms and SWRL rules.

The ontology-based approach was used in this study not only because of its logic foundation that supports logic-based reasoning but also because of its capability in achieving semantic interoperability among different information systems (as explained in section 2.1). In this approach, all related data is represented in RDF and SPARQL can be used to manipulate the RDF data for compliance checking. Recently, researches have attempted to extend SPARQL with spatial functions to enable spatial query/reasoning over RDF data. The two most notable outcomes are GeoSPARQL [37] and BimSPARQL [19]. GeoSPARQL provides a set of topological and geospatial SPARQL extensions for spatial computations in the geospatial domain. BimSPARQL provides domain-specific SPARQL extensions for querying IFC building data in applications that involve spatial reasoning. While Yurchyshyna and Zarli [32] has attempted the use of ontology and SPARQL for the conformance checking of construction projects, it is limited in formalizing the regulatory requirements that contain spatial rules. Technical challenges in automating the compliance checking of underground utilities include 1) the heterogeneity in utility data and 2) the dominance of spatial constraints in utility regulations. Towards that end, this study presents an ontology-based approach to integrate heterogeneous utility data and further adds spatial extensions to SPARQL to realize the checking of spatial utility data against spatial rules.

2.3 Proposed Semantic Approach to Compliance Checking of Underground Utilities

Figure 2.2 illustrates the overall framework created in this study to support utility compliance checking. With a focus on semantics, the proposed approach employs ontology to integrate heterogeneous data and enables automated compliance checking through logic and spatial reasoning. The framework is composed of three main modules: ontology interlinking module, RDF conversion module, and compliance checking module, as follows.

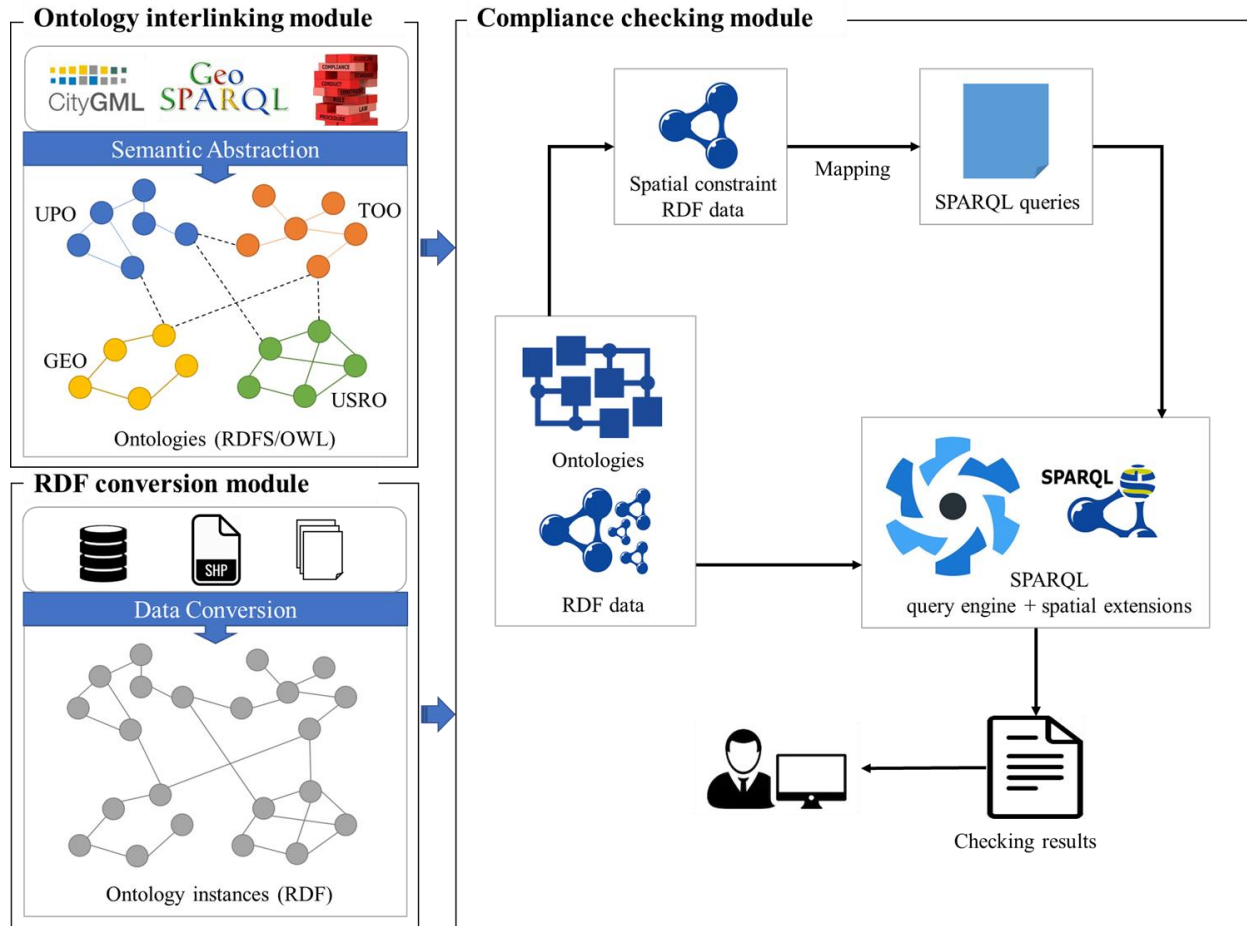


Figure 2.2. The overall framework for utility compliance checking

2.3.1 Ontology interlinking module

Four ontologies were developed and semantically linked to build the ontology-based semantic schema for heterogeneous data relevant to utility compliance checking. These four ontologies include utility product ontology (UPO), transportation object ontology (TOO), geometry ontology (GEO), and utility spatial rule ontology (USRO). They were created by abstracting semantics from open data standards including CityGML, CityGML Utility Network ADE, and GeoSPARQL, and utility regulations. They were linked by defining relationships between applicable concepts across different ontologies. The semantic schema offers a uniform platform for structuring heterogeneous geospatial and textual data of utilities and meanwhile remains linkable to other existing or newly developed ontologies for data integration from other sources. The development of ontologies is presented in section 2.4.

2.3.2 RDF conversion module

RDF was chosen in this study as the uniform and linkable data format for ontology instances, i.e., the contents of the ontologies, and RDF converters have been developed to convert heterogeneous data relevant to utility compliance checking (i.e., geospatial data of utilities and their surroundings and textual data of utility regulations) into the RDF format. RDF converters are typically platform-specific, such as the those for converting from relational data [84], IFC [28,85], and LandXML [10]. Mapping rules must be established to connect the source and target semantic schema in order to convert data into RDF format. Figure 2.3 illustrates the architecture of RDF converters. In this study, a new set of mapping rules were established to customize the two RDF converters – TripleGeo and TripleText – to transform geospatial information of urban infrastructure (usually stored in ESRI Shapefiles) and utility spatial constraint information (textual descriptions), respectively into the RDF format, following the semantic schema of the corresponding ontologies. The detailed RDF conversion process is presented in section 2.5.

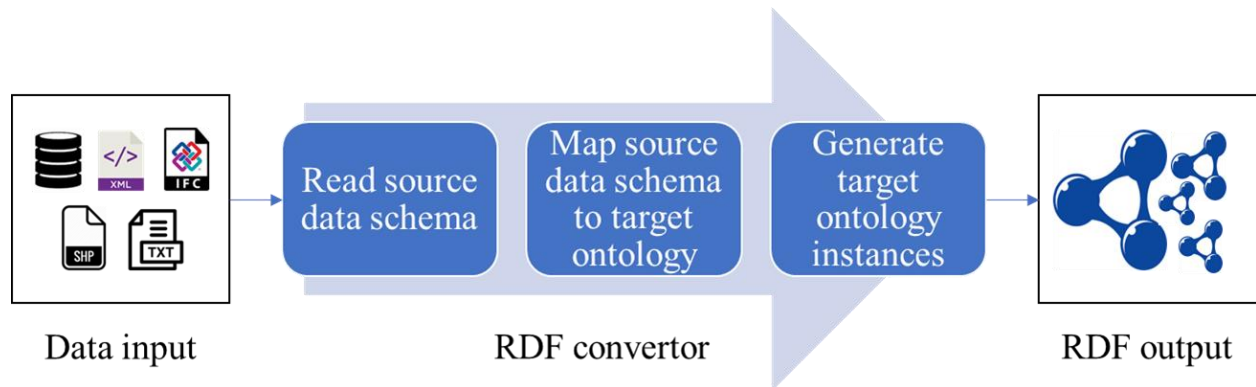


Figure 2.3. The architecture of RDF converters

2.3.3 Compliance checking module

Since RDF is the uniform data format for heterogeneous data relevant to utility compliance checking, SPARQL, an RDF query language, was used in this module. The compliance checking is performed by SPARQL queries. The overall process was designed as a series of SPARQL-based RDF data manipulations: retrieval of spatial constraints, generation of SPARQL queries, execution of SPARQL queries, and reporting. To generate expressive SPARQL queries from spatial constraint RDF data for compliance checking, two key technical components were used: 1) a mechanism for mapping semantic-related RDF data (such as the mapping between utility product

data and regulation data) and 2) a list of SPARQL spatial extensions for spatial manipulation over RDF data. Section 2.6 presents more details about the compliance checking process.

2.4 Ontology Development and Interlinking

Ontology development methodologies have been suggested by several authors such as El-Gohary and El-Diraby [86]. Although there is variation among these methods, they all include five key steps: (1) purpose and scope definition, (2) taxonomy building, (3) relation modeling, (4) ontology coding, and (5) ontology evaluation. Following the five-step procedure, four ontologies (i.e., UPO, TOO, GEO, and USRO) were developed to provide the semantic schema to integrate heterogeneous utility data for the purpose of compliance checking. The scopes of these ontologies were determined by developing a set of competency questions [38] that the ontology should be able to answer. In this study, a total of 36 competency questions were designed to capture ontology engineering requirements. Examples of competency questions include:

- Which sector does utility pipe X belong to?
- What are the dimensions (such as length, diameter, thickness, etc.) of water pipe X?
- What type of surface material does driving lane X have?
- What is the location of sewer manhole X?
- What is the spatial constraint to water pipe X?
- What are the constrained urban products in spatial constraint X?

For taxonomy building, the main concepts/classes in the domain of interest were identified based on a review of relevant open standards and textual documents. The concept/class taxonomy was built up following the top-down (starting by defining the most abstract concepts) approach. For relation modeling, relationships between concepts were identified to provide detailed information about the defined concepts. Specifically, UPO and TOO provide the conceptualization of urban infrastructure products (i.e., utilities and roads) with non-spatial/thematic properties. GEO provides geometry-relevant concepts to capture the geometry and location of urban infrastructure products. USRO models the semantics behind the textual descriptions of spatial constraints and rules in utility regulations and specifications. The cross-ontology linkage was also established by defining relationships between concepts across different ontologies. The developed ontologies were encoded in OWL format. The evaluation of the developed ontologies was conducted in an

iterative manner, starting with evaluation through simple automated consistency checking to ensure correct syntax formalization (using the built-in Protégé reasoner), followed by conformance checking to the set of predefined competency questions (through SPARQL queries), and finally evaluation by domain experts. The developed ontologies were evaluated to be accurate, sufficient and shared conceptualizations of the related domains.

2.4.1 UPO

UPO was built based on the semantics of CityGML Utility Network ADE. Figure 2.4 illustrates its resulting ontology structure with eight classes and eight properties defined. Following the semantic classification of utility network products provided in CityGML Utility Network ADE, the class of `utilityProduct` is the central concept of UPO, representing general utility products and is further specialized into five subclasses: `distributionElement`, `functionalElement`, `protectiveElement`, `terminalElement`, and `device`. Thematic attributes such as dimensions and material types are captured as specific datatype properties of `utilityProduct`. For instance, the datatype property of `hasDiameter` is defined to hold the diameter value of utility products. Additional properties (e.g., `belongsToSector`, `encasedBy`) are also introduced in UPO to establish domain-specific relationships among `utilityProduct` and relevant classes (e.g., `utilitySector` and `encasement`). For the `utilitySector` class, the list of instances includes water, sewer, stormwater, gas, electricity, and telecom. This list enables the sector-characterization of utility products. Geometric attributes of utility products are captured in a separate ontology (see section 2.4.3).

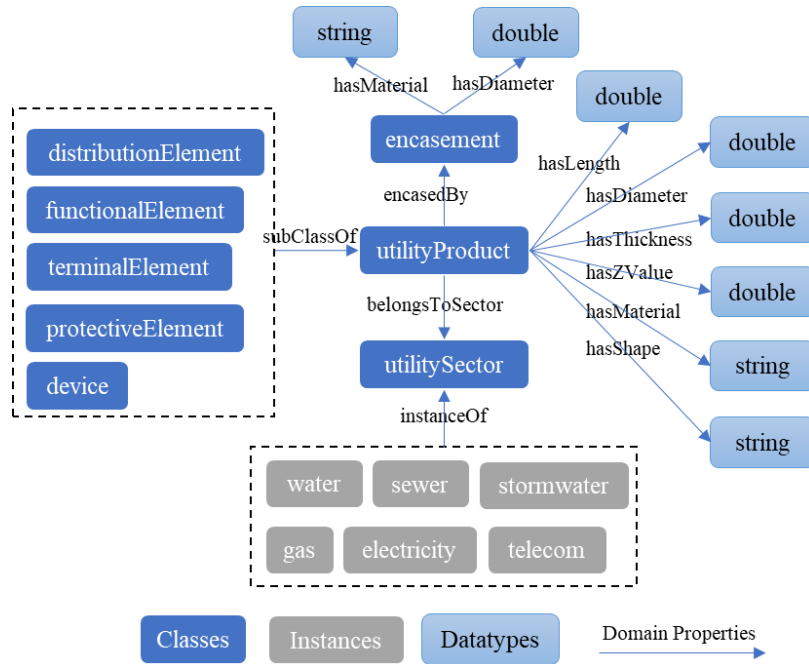


Figure 2.4. Utility product ontology

2.4.2 TOO

TOO was built based on the semantics of the CityGML transportation model. Figure 2.5 illustrates its ontology structure with eleven classes and eight properties defined. In CityGML, the transportation objects are defined by classes related to geometric primitives (such as points, lines, and polygons) and non-geometric attributes (such as dimensions and surface material types). The main class is the transportation complex, which represents, for example, a road, a track, a railway, or a square. A transportation complex is composed of the parts: traffic area and auxiliary traffic area. Following the semantics of CityGML, in the newly created TOO transportationObject is the main class for representing the general transportation objects. It is composed of auxiliary and regular traffic areas, represented as trafficArea and auxiliaryTrafficArea classes, respectively. The isComposedOf property links transportationObject, trafficArea and auxiliaryTrafficArea classes. The subClassOf property is used to further specialize trafficArea into drivingLane, cyclepath, footpath, and intersection, and auxiliaryTrafficArea into ditch, embankment, trafficIsland, and shoulder. Same as UPO, non-geometric attributes are defined as datatype properties associated with transportation objects while geometric attributes are captured by linking to the separate GEO (described in section 2.4.3).

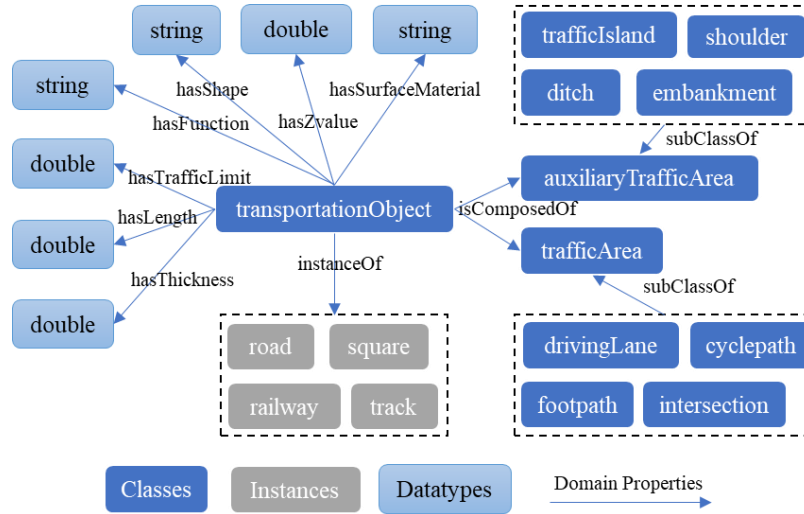


Figure 2.5. Transportation object ontology

2.4.3 GEO

OGC developed the GeoSPARQL standard to define a vocabulary for representing features, geometries, and their relationships. This study adopts the definitions of geometry-related concepts from GeoSPARQL and formulates GEO for representing the geometric attributes of urban infrastructure products. Figure 2.6 illustrates the ontology structure of GEO with eight classes and four properties defined. In GEO, the class `spatialObject` is defined with two primary subclasses, `feature` and `geometry`. `feature` represents physical objects while `geometry` represents geometry objects. They are linked via the `hasGeometry` property: `feature` – `hasGeometry` – `geometry`. Typical geometry primitives such as `point`, `line`, and `polygon` are also given as subclasses of `geometry` to represent geometric details. In addition, GEO includes two different ways to represent geometry literals: Well Known Text (WKT) and Geography Markup Language (GML), linked to `geometry` via the `asWKT` and `asGML` properties respectively. Within the class of `spatialObject`, the property `hasSpatialRelation` was added to describe the spatial configurations between spatial objects. In GeoSPARQL, a set of topological relations (such as `geo:sfIntersects`, `geo:sfWithin`, `geo:sfDisjoint`, etc.) are provided with the capability of spatial reasoning, used to ask for certain spatial relationships between spatial objects. They were defined as sub properties of the `hasSpatialRelation` property. Regarding GeoSPARQL geospatial functions (such as `geof:distance`), this study also includes them as GEO spatial extensions.

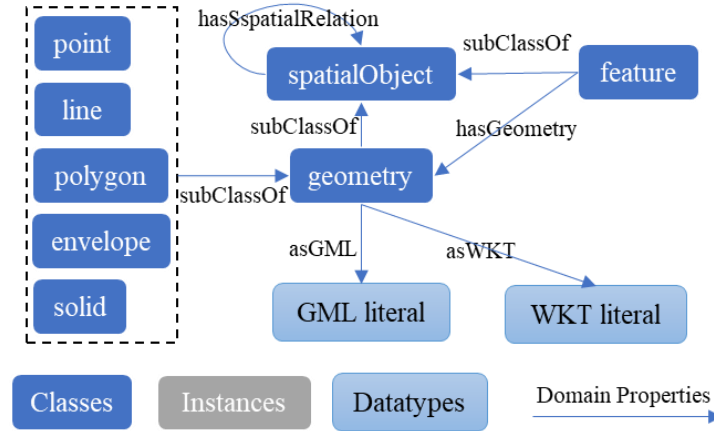


Figure 2.6. Geometry ontology

2.4.4 USRO

Utility compliance checking requires the interaction with textual data of utility regulations. USRO is developed in this study to capture the semantics behind the spatial cognitive-linguistic expressions used in utility regulations at the sentence level. Figure 2.7 illustrates the ontology structure of USRO with two classes and twelve properties defined. The head class of this ontology is *spatialConstraint*, which refers to the original textual description of a spatial constraint (e.g., all pipelines must have a minimum depth of cover of 4 feet under ditches). A spatial constraint may specify multiple spatial configurations among entities using conjunctions (e.g., and). The *spatialConfiguration* class is therefore connected to *spatialConstraint* via the *specifies* property and the *inConjunctionWith* property is defined to indicate the logical connectives between *spatialConfiguration*. Under the *inConjunctionWith* property, there are two specific sub properties: *conjunctionAnd* and *conjunctionOr*. For those natural language expressions that are used to describe the specified spatial configurations, USRO models them as datatype properties. Specifically, the *hasTrajectory* property holds the string value of the central object (e.g., pipeline) of the spatial configuration; the *hasTrajectoryAttribute* property is used to describe the attribute of the trajectory; the *hasSpatialIndicator* property holds a word or a phrase (e.g., under, depth of cover) for a spatial relation between spatial objects; the *hasLandmark* property holds the string value (e.g., ditch) of a secondary object of the spatial configuration, to which a possible spatial relation can be established; the *hasLandmarkAttribute* property is used to describe the attribute of the landmark; the *hasDistanceValue* and *hasDistanceUnit* properties capture the value (e.g., 4) and unit (e.g., feet)

of the distance between a trajector and a landmark; the `hasDistanceRestriction` property refers to the restriction set to the distance value (e.g., minimum); the `hasNegation` property is used to describe the existence of the word “no” or “not” in the sentence; and the `hasRequirementIndicator` property holds the word or phrase (e.g., must) that indicates a requirement type: obligation, permission, or prohibition.

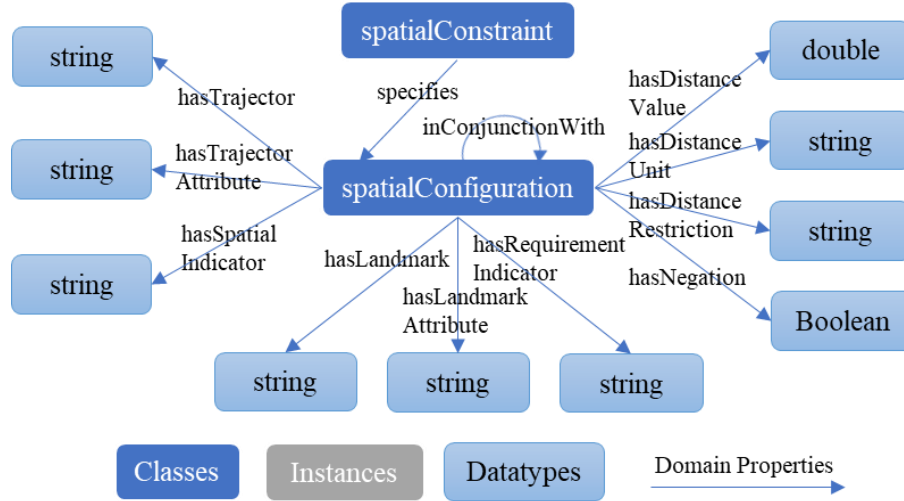


Figure 2.7. Utility spatial rule ontology

2.4.5 Cross-ontology linkage

To build the semantic framework for utility compliance checking, four developed ontologies are linked by defining relationships between applicable concepts across different ontologies. As in Figure 2.8, `utilityProduct` – `subClassOf` – `feature` and `transportationObject` – `subClassOf` – `feature` links are established to enable the integration of urban product data and geometry data. The location-related interactions between utilities and their surroundings are captured via `utilityProduct` – `interactsWith` – `transportationObject`. The mappings from utility product data and the text mentions in regulations are established by `spatialConstraint` – `constrains` – `utilityProduct`. In addition, the utilities meeting/not meeting the regulatory requirements can be linked to specific requirements via the `isCompliantWith/isNonCompliantWith` property. The four interlinked ontologies provide the ontological framework for supporting utility compliance checking.

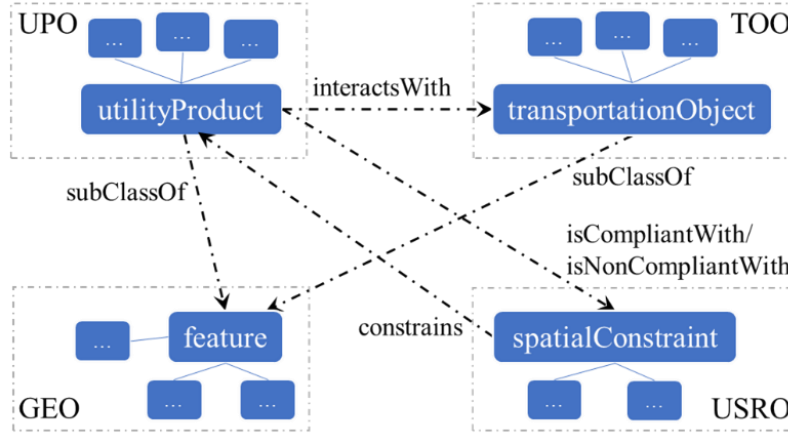


Figure 2.8. Cross-ontology linkage

2.5 RDF Data Conversion

RDF converters were customized/developed in this study to convert heterogeneous data (i.e., geospatial data of utilities and their surroundings and textual data of utility regulations) into the RDF format. This section presents the conversion process in detail.

2.5.1 RDF convertor for geospatial data

The geospatial information of urban infrastructure is typically modeled in GIS formats, such as standard geographic format (e.g., Shapefile) or widely used spatial database format (e.g., Oracle Spatial or PostGIS). TripleGeo [67], an open source conversion tool, provides the capability of accessing both thematic attributes and geometric representations from GIS and transforming them into RDF and the geometry vocabulary prescribed by GeoSPARQL. In this study, TripleGeo was extended to convert the geospatial information of transportation infrastructure and underground utilities from Shapefile to UPO, TOO, and GEO-compliant RDF.

Figure 2.9 illustrates the working process. First, the target ontology is specified (e.g., UPO and TOO) and the source Shapefile is loaded. The convertor reads the Shapefile and displays its attribute table and the classes/properties (of the target ontologies). Then, each attribute in the attribute table is connected to its correspondence in the ontologies (see the mapping tables in Figure 2.9).

TripeGeo

Loaded ontologies: Loaded Shapefiles:

Shapefile attribute table preview

	OBJECTID *	SHAPE *	FacilityID	UtilSector	UtilType	FuncType	UtilMat	UtilDia	FacShape	Encas	BasElev	SHAPE_Length
▶	1	Polyline	WM4110	Water	Water Main	Distribution	DIP	12	Round	Steel	695.25	487.530146
	2	Polyline	WM2516	Water	Water Main	Distribution	CAS	8	Round	Steel	695.75	636.115859
	3	Polyline	WM1799	Water	Water Main	Distribution	CAS	8	Round	Steel	695.75	65.318096
	4	Polyline	WL11249	Water	Water Lateral	Distribution	CAS	8	Round	<Null>	695.75	66.070011
	5	Polyline	WL4228	Water	Water Lateral	Distribution	PVC	12	Round	<Null>	695.75	69.142344
	6	Polyline	WL1611	Water	Water Lateral	Distribution	PVC	12	Round	<Null>	695.75	63.445051
	7	Polyline	WL2632	Water	Water Lateral	Distribution	CAS	8	Round	<Null>	695.75	62.678401
	8	Polyline	WL1516	Water	Water Lateral	Distribution	CAS	8	Round	<Null>	695.75	65.295879
	9	Polyline	WL1161	Water	Water Lateral	Distribution	PVC	12	Round	<Null>	695.75	66.461547

Mapping Tables

Thematic attributes mapping			Polyline Geometry mapping		
Attribute Field	Data Type	Ontology Property	Field Name	Data Type	Ontology Property
OBJECTID	Object ID (Auto)	N/A	SHAPE	Geometry	geo:hasGeometry
UtilSector	Text	upo:belongsToSector	START_X	Double	asWKT
UtilType	Text	rdfs:label	START_Y	Double	asWKT
FuncType	Text	rdf:type	END_X	Double	asWKT
UtilMat	Text	upo:hasMaterial	END_Y	Double	asWKT
UtilDia	Double	upo:hasDiameter			
FacShape	Text	upo:hasShape			
Encas	Text	upo:encasedBy/upo:h...			
BasElev	Double	upo:hasZValue			
SHAPE_Length	Double	upo:hasLength			

Target Geo-Vocabulary

Figure 2.9. Conversion from Shapefile to RDF

Specifically, the convertor transforms data into RDF format by completing three tasks: 1) processing thematic attributes, 2) processing geometric attributes, and 3) linking UPO/TOO and GEO instances. For the thematic attributes, the process of converting Shapefile data to RDF is initiated by creating an instance of an ontology class for each row in the Shapefile attribute tables, followed by adding properties to the instance based on the mappings. For instance, in Figure 2.9, for each row of the previewed Shapefile attribute table, an ontology instance of the UPO class `utilityProduct` is generated based on the `FacilityID` field. The `UtilSector` field indicates a `belongsToSector` object property added to the `utilityProduct` instance. The `UtilType` field indicates a `rdfs:label` string property to describe the human-readable version of the instance name. The `FuncType` field indicates that the generated instance is a type of the `distributionElement` class. The other attributes of `UtilMat`, `UtilDia`, `FacShape` and `BasElev` are linked to the instance via properties of `hasMaterial`, `hasDiameter`, `hasShape` and `hasZValue`. The `Encas` attribute is quite unique as it

stores information relevant to the UPO triple of utilityProduct – encasedBy – encasement. Consequently, an instance of the UPO encasement class is created and linked to the UtilityProduct instance via the encasedBy object property, and the encasement material attribute is described via the hasMaterial property. For geometric attributes, GEO geometry instances (one for each row in the attribute table) are first generated for the SHAPE geometry field. The geometric representations accessed from TripleGeo are then added as GEO properties to the geometry instances in GEO-compliant RDF triples. For instance, the parsed coordinates of the starting and ending points of the polyline geometry are serialized as WKT literals and linked to the geometry instance via the asWKT property. Finally, the UPO/TOO instances (also established as GEO feature instances) are linked to GEO instances via the hasGeometry property to achieve the integration of urban product data and geometry data.

While Figure 2.9 illustrates the process using a utility file in the Shapefile format as the example, the same process applies to GIS data of utility and transportation from databases such as Oracle Spatial and Post GIS.

Figure 2.10 provides an excerpt of the resulting RDF output in Turtle, which is a common format for storing RDF data in textual representations.

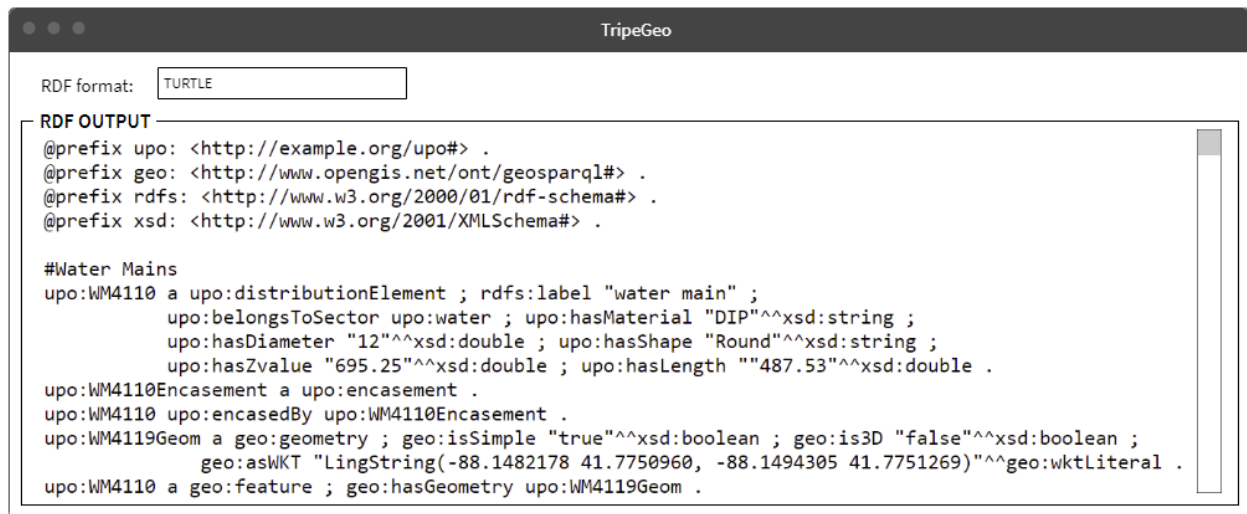


Figure 2.10. An excerpt of the resulting RDF output in Turtle format

2.5.2 RDF convertor for textual data

Attributed to the overwhelming role of location and clearances in achieving most decision criteria and the dominance of such spatial configuration-related requirements predominate in

utility regulatory documents [2], textual data processed in this study are the sentences that contain spatial constraints. Figure 2.11 gives a collection of examples of utility spatial constraints.

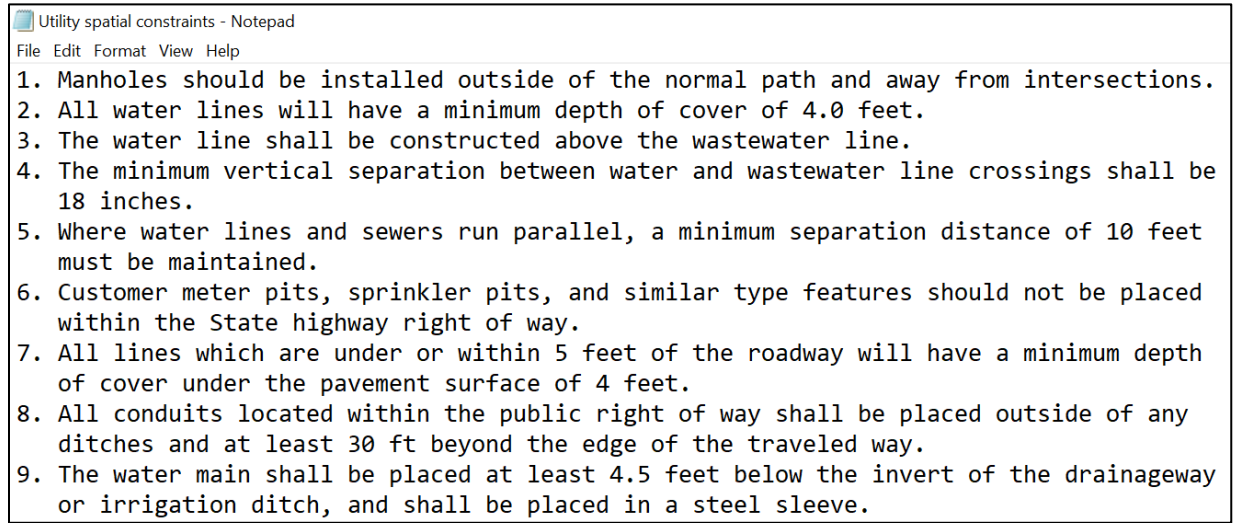


Figure 2.11. Examples of utility spatial constraints

Utility spatial constraints are usually described using natural language texts. The NLP algorithm designed by Li et al. [1] was implemented in this study to extract all required spatial cognitive-linguistic elements from the textual spatial constraints and present them as 10-element tuples of <Trajector, Trajector attribute, Spatial indicator, Landmark, Landmark attribute, Requirement indicator, Negation, Distance value, Distance unit, Distance restriction>. One 10-tuple is the structured representation of one spatial configuration specified in the spatial constraints. There could be multiple spatial configurations in a sentence. The NLP-based extraction results were evaluated by testing 30 sentences obtained from utility regulations [87,88]. It was found that for sentences containing no more than two spatial configurations, the extraction is 100% accurate. However, for sentences that contain more than two spatial configurations, only 74.24% accuracy was achieved and errors such as missing landmarks and incorrect spatial indicators exist. Manual adjustments are required in order to achieve the extraction accuracy for subsequent RDF conversion. Figure 2.12 illustrates the three resulting 10-tuples from two sentences that contain spatial constraints.

Extraction of spatial cognitive-linguistic elements																																																																					
Sample sentences of spatial constraints Utility spatial constraints - Notepad File Edit Format View Help 1. Manholes should be installed outside of the normal wheel path and away from intersections. 2. All water lines will have a minimum depth of cover of 4.0 feet.																																																																					
Cognitive-linguistic elements in tuples <table> <tr> <th rowspan="2">Spatial constraint No.</th><th rowspan="2">Spatial configuration No.</th><th colspan="10">Cognitive-linguistic elements in tuples</th></tr> <tr> <th>Trajector</th><th>Trajector attribute</th><th>Spatial indicator</th><th>Landmark</th><th>Landmark attribute</th><th>Requirement indicator</th><th>Negation</th><th>Distance value</th><th>Distance unit</th><th>Distance restriction</th></tr> <tr> <td>1</td><td>1</td><td>manhole</td><td>N/A</td><td>outside of</td><td>wheel path</td><td>N/A</td><td>should</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>1</td><td>2</td><td>manhole</td><td>N/A</td><td>away from</td><td>intersection</td><td>N/A</td><td>should</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>2</td><td>1</td><td>water line</td><td>N/A</td><td>depth of cover</td><td>roadway (implicit)</td><td>N/A</td><td>will</td><td>N/A</td><td>4.0</td><td>foot</td><td>minimum</td></tr> </table>												Spatial constraint No.	Spatial configuration No.	Cognitive-linguistic elements in tuples										Trajector	Trajector attribute	Spatial indicator	Landmark	Landmark attribute	Requirement indicator	Negation	Distance value	Distance unit	Distance restriction	1	1	manhole	N/A	outside of	wheel path	N/A	should	N/A	N/A	N/A	N/A	1	2	manhole	N/A	away from	intersection	N/A	should	N/A	N/A	N/A	N/A	2	1	water line	N/A	depth of cover	roadway (implicit)	N/A	will	N/A	4.0	foot	minimum
Spatial constraint No.	Spatial configuration No.	Cognitive-linguistic elements in tuples																																																																			
		Trajector	Trajector attribute	Spatial indicator	Landmark	Landmark attribute	Requirement indicator	Negation	Distance value	Distance unit	Distance restriction																																																										
1	1	manhole	N/A	outside of	wheel path	N/A	should	N/A	N/A	N/A	N/A																																																										
1	2	manhole	N/A	away from	intersection	N/A	should	N/A	N/A	N/A	N/A																																																										
2	1	water line	N/A	depth of cover	roadway (implicit)	N/A	will	N/A	4.0	foot	minimum																																																										

Figure 2.12. Extraction of spatial cognitive-linguistic elements from spatial constraint sentences

An RDF convertor entitled TripleText was developed in this study to convert the extracted spatial rules into the RDF format, following the semantic structure of USRO. Figure 2.13 illustrates the mapping process of TripleText. Starting from each row in the table of spatial rules, it creates a USRO spatialConstraint instance for each spatial constraint (e.g., spatialConstraint1) and a USRO spatialConfiguration instance for each spatial configuration (e.g., spatialConfiguration11), and links spatialConfiguration instances to their corresponding spatialConstraint instance via the specifies property (e.g., spatialConstraint1 – specifies – spatialConfiguration11). The inConjunctionWith property between spatialConstraint instances are specified based on the conjunctions used in the original sentences. For example, in spatial constraint No. 1, two described spatial configurations are connected using the conjunction “and”, thus an RDF triple of spatialConfiguration11 – conjunctionAnd – spatialConfiguration12 was established. For the extracted spatial cognitive-linguistic elements, each of the ten elements in a tuple leads to a USRO datatype property that is connected to the corresponding spatialConfiguration instance. For example, the landmark element maps to the string property of hasLandmark, established such as spatialConfiguration11 – hasLandmark – “wheel path”. In such a way, utility spatial constraints (represented as ten-element tuples) can be converted into USRO-compliant RDF instances.

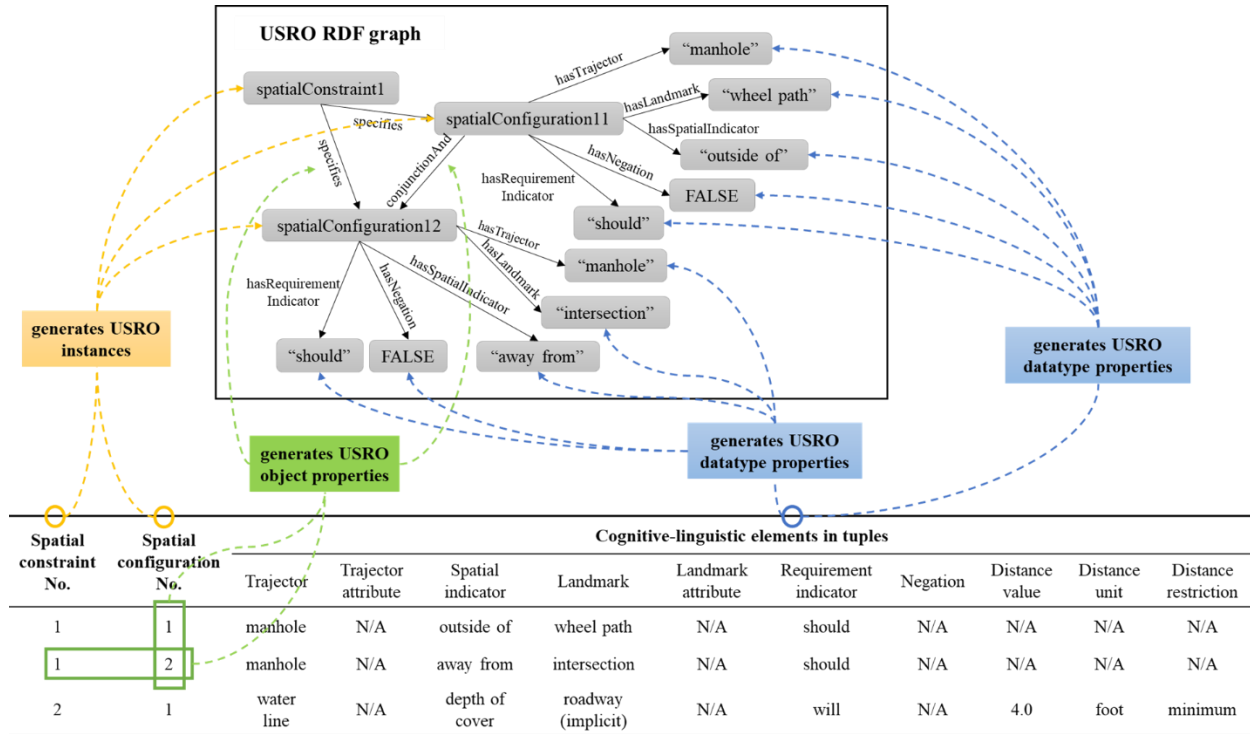


Figure 2.13. Mapping process of TripleText

Figure 2.14 illustrates the resulting RDF data for the two spatial configurations under spatial constraint No. 1.

```

TripleText
RDF output (Turtle format)
@prefix usro: <http://example.org/usro#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

usro:spatialConstraint1 a usro:spatialConstraint ;
    rdfs:label "Manholes should be installed outside
    the normal wheel path and away from intersections" ;
    usro:specifies usro:spatialConfiguration11 ;
    usro:specifies usro:spatialConfiguration12 .

usro:spatialConfiguration11 a usro:spatialConfiguration ;
    usro:hasTrajectory "manhole"^^xsd:string ;
    usro:hasSpatialIndicator "outside of"^^xsd:string ;
    usro:hasLandmark "wheel path"^^xsd:string ;
    usro:hasRequirementIndicator "should"^^xsd:string ;
    usro:hasNegation "false"^^xsd:boolean .

usro:spatialConfiguration12 a usro:spatialConfiguration ;
    usro:hasTrajectory "manhole"^^xsd:string ;
    usro:hasSpatialIndicator "away from"^^xsd:string ;
    usro:hasLandmark "intersection"^^xsd:string ;
    usro:hasRequirementIndicator "should"^^xsd:string ;
    usro:hasNegation "false"^^xsd:boolean .

usro:spatialConfiguration11 usro:conjunctionAnd usro:spatialConfiguration12 .

```

Figure 2.14. An excerpt of the resulting RDF output in Turtle format

2.6 Utility Compliance Checking

The purpose of the compliance checking module is to retrieve the spatial constraints from USRO and check the utility data (stored in UPO) and transportation infrastructure data (stored in TOO). The compliance checking is performed by SPARQL queries. Three tasks are involved to generate SPARQL queries: retrieving spatial constraints from USRO, mapping between USRO and UPO/TOO RDF data, and extensions to SPARQL.

(1) Retrieval of spatial constraints from USRO

USRO contains all the spatial constraints and their spatial configurations in the RDF format. SPARQL queries have been developed to retrieve specific data elements of spatial configurations for each spatial constraint. For instance, the SPARQL query in Figure 2.15 returns detailed information for spatial configurations under spatialConstraint1: manholes should be installed outside the normal wheel path and away from intersections.

```
SELECT ?spatialConfiguration ?trajector ?trajectorAttribute
      ?spatialIndicator ?landmark ?landmarkAttribute
      ?requirementIndicator ?negation ?distanceValue
      ?distanceUnit ?distanceRestriction
WHERE {
  usro:spatialConstraint1 usro:specifies ?spatialConfiguration .
  ?spatialConfiguration usro:hasTrajector ?trajector ;
                        usro:hasTrajectorAttribute ?trajectorAttribute ;
                        usro:hasSpatialIndicator ?spatialIndicator ;
                        usro:hasLandmark ?landmark ;
                        usro:hasLandmarkAttribute ?landmarkAttribute ;
                        usro:hasRequirementIndicator ?requirementIndicator ;
                        usro:hasNegation ?negation ;
                        usro:hasDistanceValue ?distanceValue ;
                        usro:hasDistanceUnit ?distanceUnit ;
                        usro:hasDistanceRestriction ?distanceRestriction .
}
GROUP BY ?spatialConfiguration
```

Figure 2.15. An example of SPARQL query for retrieval of spatial constraint information

(2) Mapping between USRO and UPO/TOO RDF data

Once specific information of every spatial configuration is retrieved, SPARQL queries for compliance are developed. A SPARQL query is an assembly of RDF triple query patterns, conjunctions, disjunctions (UNION), negations (NOT EXISTS), SPARQL functions, etc. The USRO RDF data is quite informative for formalizing these SPARQL queries.

First, the string values of USRO hasTrajectory and hasLandmark properties indicate the type of UPO/TOO product RDF triple query patterns to be constructed in the target SPARQL queries. However, the vocabulary used for an urban infrastructure product in GIS might be different from that in utility regulations. For instance, water line in utility regulations could be water pipeline in GIS. In the newly created RDF data, the `rdfs:label` literal property is used to hold the names of UPO and TOO instances while the USRO hasTrajectory and hasLandmark properties hold the string values of the constrained products in regulation. Therefore, the semantic mapping of these terminologies is necessary in order to develop SPARQL queries. A domain semantic resource that archives synonyms (is-similar), hyponyms (type-of), and meronyms (part-of) of heterogeneous terminologies often serves as the junction to facilitate the mapping process. This study adopts two existing semantic resources: the list developed by Li et al. [1] which stores semantic-related terms for the utility domain and the digital dictionary developed by Le and Jeong [37] for the transportation domain. A partial view of the semantic resource is given in Figure 2.16.

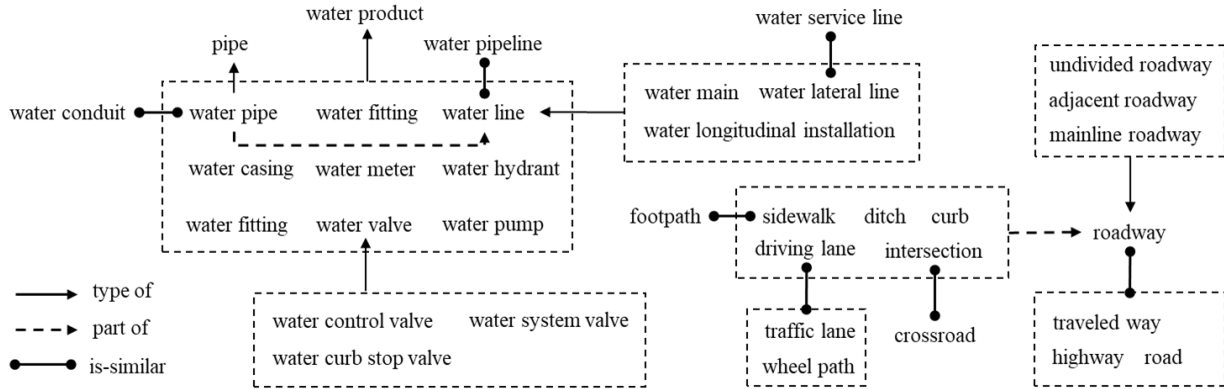


Figure 2.16. A partial view of the semantic resource for urban infrastructure domain

Terminologies used in different ontologies are matched by consulting the semantic resource. For instance, if water line is specified in the USRO, any UPO instance whose name is in the synonym set or the hyponym set of water line will be checked. The corresponding SPARQL query is illustrated in Figure 2.17.

```

SELECT ?waterLine
WHERE {
  ?waterLine a upo:utilityProduct;
             rdfs:label ?mappedTerm1.
  FILTER (?mappedTerm1 = "water line" || ?mappedTerm1 = "water pipeline"
  || ?mappedTerm1 = "water main" || ?mappedTerm1 = "water lateral line"
  || ?mappedTerm1 = "water service line").
}

```

Figure 2.17. The SPARQL query for selecting the utility product of water line

Second, the USRO hasTrajectoryAttribute and hasLandmarkAttribute properties indicate the type of UPO/TOO product-attribute RDF triple query patterns to be constructed in the target SPARQL queries. Based on this, product attributive information captured in USRO needs to be mapped to the designated UPO/TOO attributes. For example, a trajectory attribute of 6-inch in diameter maps to the UPO hasDiameter property while a landmark attribute of high-speed (exceeding 50 mph) maps to the TOO hasTrafficLimit property for constructing UPO/TOO product-attribute RDF triple patterns.

Third, the USRO hasSpatialIndicator properties indicate the type of SPARQL functions to be used in the target SPARQL queries. Based on this, SPARQL needs to be extended as functions for direct use in the target queries and spatial indicators in USRO need to be mapped to the SPARQL functions for specific uses.

Finally, the conjunctions, disjunctions, and negations among the mapped UPO/TOO RDF triples and the specific uses of the mapped SPARQL functions can be determined based on the USRO requirement indicators, negation information, distance information, and the specific conjunctions between the spatial configurations.

(3) Spatial extensions to SPARQL

Based on the spatial indicators in USRO, this task aims to add new case-specific spatial extensions to SPARQL. SPARQL queries use spatial functions for spatial reasoning over RDF data. While GeoSPARQL topological and geospatial functions are included in GEO as spatial extensions, these functions are mainly for 2D geometry with very limited capacity for spatial reasoning in 3D. In this study, four new functions were created and implemented through SPARQL Inferencing Notation (SPIN) [89] as spin:depthOfCover, spin:verticalDistance, spin:below, and spin:above. The spin:depthOfCover function computes the buried depth of a utility product under

a transportation object; the `spin:verticalDistance` function computes the vertical distance between two utility products based on their elevations; the `spin:below` and `spin:above` functions evaluate the vertical directional relationships (below or above) between two objects based on their elevations. Code excerpts in Figure 2.18 provide details regarding the implementation of these four new functions in SPIN.

```
# spin:depthOfCover
SELECT ?depthOfCover
WHERE {
  ?argument1 a upo:utilityProduct ;
              upo:hasZValue ?value1 .
  ?argument2 a too:transportationObject ;
              too:hasZValue ?value2 .
  BIND (abs(?value1 - ?value2) AS ?depthOfCover) .
}

# spin:verticalDistance
SELECT ?verticalDistance
WHERE {
  ?argument1 a upo:utilityProduct ;
              upo:hasZValue ?value1 .
  ?argument2 a upo:utilityProduct ;
              upo:hasZValue ?value2 .
  BIND (abs(?value1 - ?value2) AS ?verticalDistance) .
}

# spin:below
SELECT ?rightSideArgument
WHERE {
  ?leftSideArgument upo:hasZValue|too:hasZValue ?value1 .
  ?rightSideArgument upo:hasZValue|too:hasZValue ?value2 .
  FILTER (?value1 < ?value2) .
}

# spin:above
SELECT ?rightSideArgument
WHERE {
  ?leftSideArgument upo:hasZValue|too:hasZValue ?value1 .
  ?rightSideArgument upo:hasZValue|too:hasZValue ?value2 .
  FILTER (?value1 > ?value2) .
}
```

Figure 2.18. Code excerpts of extended SPARQL functions using SPIN

USRO accommodates a variety of spatial indicators, terms relevant to spatial relationships. Table 2.1 illustrates the mapping from the indicators of spatial relationships in USRO to these four new functions and relevant functions from GeoSPARQL.

Table 2.1. Mappings from spatial indicators to spatial functions

Spatial indicators in USRO	Mapped SPARQL extensions
above, over	spin:above
under, below	spin:below
vertical separation, vertical clearance, vertical buffer	spin:verticalDistance
depth of cover, depth of bury, buried depth	spin:depthOfCover
horizontal separation, horizontal clearance	geof:distance
crossing, intersect	geo:sfIntersects
away from, outside, outside of	geo:sfDisjoint
within	geo:sfWithin

(4) An illustrative example

The spatial constraint of “manholes should be installed outside the normal wheel path and away from intersections” is used to illustrate the generation of the SPARQL query to check UPO and TOO instances. This spatial constraint has two spatial configurations: “manhole outside wheel path” and “manhole away from intersections.” Figure 2.19 illustrates the resulting SPARQL query where the relevant products are retrieved based on the domain semantic source through the FILTER blocks. Referring to Table 2.1, the spatial indicators of “outside of” and “away from” evoke the `geo:sfDisjoint` function. Since the `geo:sfDisjoint` function works with geometry instances as its function arguments, the `hasGeometry` property of UPO and TOO instances is used to retrieve their corresponding GEO instances. To meet the spatial constraint, a manhole must be disjoint from wheel paths and intersections. When checking a manhole, NOT EXISTS, which represents a way of negation to test whether a triple exists in the triple store, is used to identify those violations. For example, NOT EXISTS {?MHGeo geo:sfDisjoint ?INGeo.} matches the manhole that is not disjoint from intersections.

```

# Manholes should be installed outside the normal wheel path and away from intersections
SELECT ?manhole
WHERE {
  ?manhole a upo:utilityProduct;
    rdfs:label ?mappedTerm1.
  FILTER (?mappedTerm1 = "sewer manhole" || ?mappedTerm1 = "storm manhole").
  ?manhole geo:hasGeometry ?MHGeo.
  ?wheelPath a too:transportationObject;
    rdfs:label ?mappedTerm2.
  FILTER (?mappedTerm2 = "wheel path" || ?mappedTerm2 = "traffic lane" || ?mappedTerm2 = "driving lane").
  ?wheelPth geo:hasGeometry ?WPGeo.
  ?intersection a too:transportationObject;
    rdfs:label ?mappedTerm3.
  FILTER (?mappedTerm3 = "intersection" || ?mappedTerm3 = "crossroad").
  ?intersection geo:hasGeometry ?INGeo.
  NOT EXISTS { ?MHGeo geo:sfDisjoint ?WPGeo.}
  UNION
  NOT EXISTS { ?MHGeo geo:sfDisjoint ?INGeo.}
}

```

Figure 2.19. An illustrative example of generated SPARQL queries

Executing the above SPARQL query detects all non-compliant manholes, or manholes that violate the constraint. Using the SPARQL CONSTRUCT method, the results of compliance checking are stored as relationships between checked objects and applicable spatial constraints for reporting and future retrieval. Specifically, the *isCompliantWith* relationship will be established between all compliant utility products and the spatial constraint while the *isNonCompliantWith* relationship will be established between all noncompliant utility products and the corresponding spatial constraints.

2.7 Implementation Architecture

Figure 2.20 illustrates the implementation architecture of the proposed approach and its data flow. Protégé was used to build the four interlinked ontologies and encode them in RDFS/OWL. Two RDF convertors – TripleGeo and TripleText – were used to convert the geospatial data of urban infrastructure and the textual data of utility regulations into RDF format as ontology instances. The RDF triple store was used to store the RDFS/OWL ontologies and their corresponding RDF instances. All the RDF data was then manipulated through the Apache Jena platform, an open source Java framework for building semantic web applications, for compliance

checking. Jena ARQ, a SPARQL-compliant query engine, was used to support SPARQL queries over RDF data. Two Jena APIs – GeoSPARQL API and SPIN API – enable the direct usage of extended spatial functions within SPARQL queries. The SPARQL query interface accepts input SPARQL queries and returns query results. Through this interface, users can 1) retrieve spatial constraint RDF data to formulate SPARQL queries, 2) execute SPARQL queries for compliance checking, 3) construct new RDF data for storage, and 4) output compliance checking results.

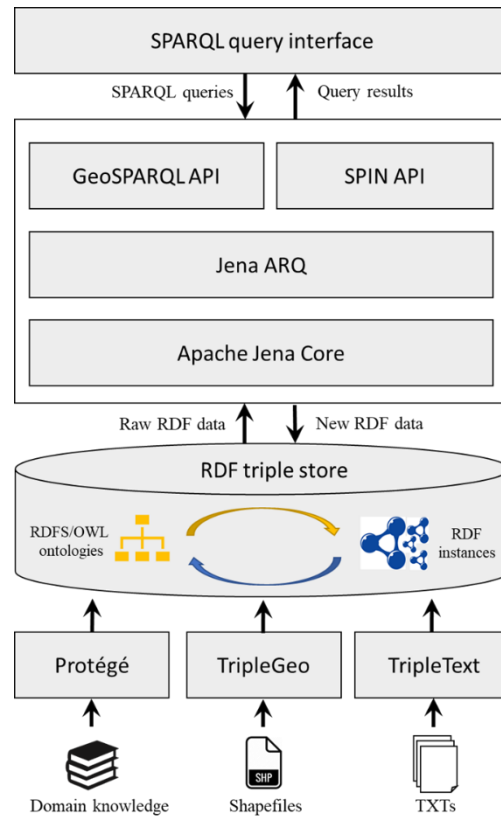


Figure 2.20. Implementation architecture and data flow

2.8 Case Illustration

The newly developed compliance checking system was tested using a sample database that is freely downloadable from ESRI.COM [90]. This database contains records for water, sewer, and stormwater infrastructure in the City of Naperville, Illinois, in GIS format. It has 62 feature classes such as water mains, storm casings, and sewer manholes, organized under three feature datasets (i.e., water distribution system, stormwater network, and sewer collection system). In this case illustration, an area of interest (AOI) was randomly chosen to test out the newly developed method for compliance checking. The resulting dataset contains a total of 212 records of pipes, casings,

valves, fittings, and other utility products. The reference street map from OpenStreetMap [91] was used to generate the transportation infrastructure feature dataset in the AOI. The digitization of these features was based on the polygon features of the reference street map. Digitized transportation feature classes include driving lanes, sidewalks, curbs, and traffic intersections. The national elevation dataset (NED) [92] was used to correct and update the elevation information for urban infrastructure elements in the AOI. Figure 2.21 illustrates the AOI and urban infrastructure data in the GIS format.

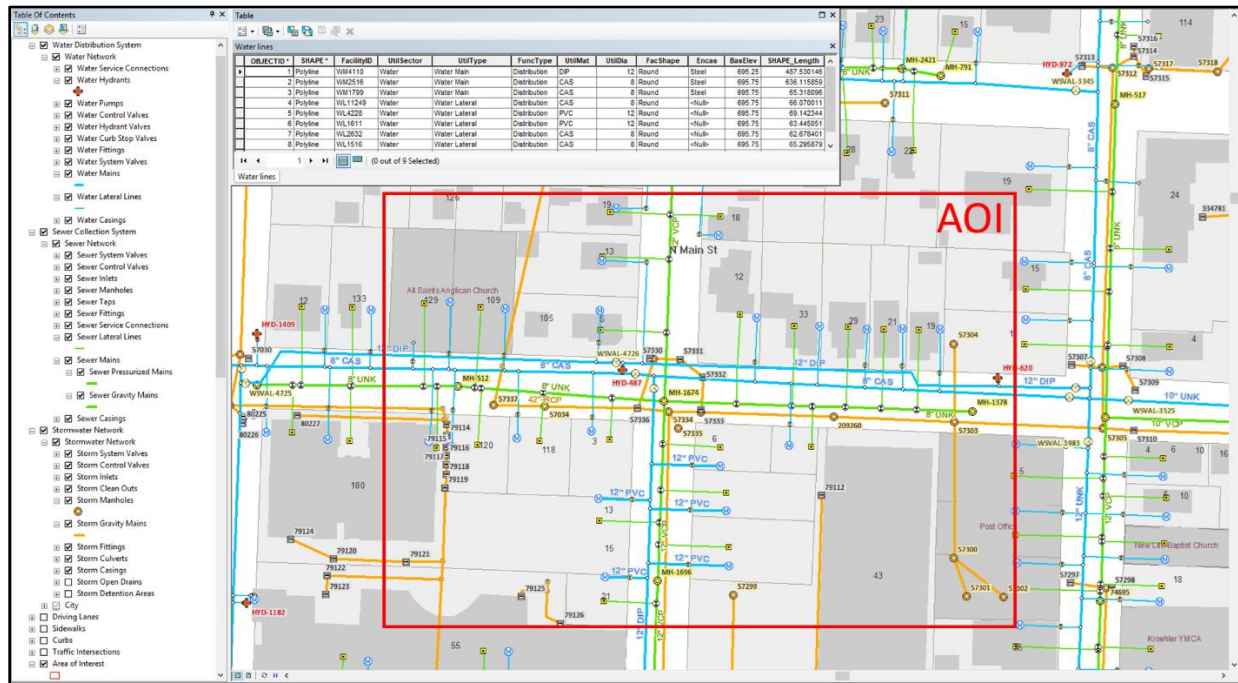


Figure 2.21. A map view of urban infrastructure in the AOI

A collection of sentences that contain spatial rules were excerpted from utility regulations [87,88] to check the compliance of the utility records. Table 2.2 provides a list of these selected constraints.

Table 2.2. The sentences of spatial constraints

No.	Spatial constraints	Constraint purposes
1	Manholes should be installed outside the normal wheel path and away from intersections	Traffic impact /Maintainability
2	All water lines will have a minimal depth of cover of 4 feet.	Infrastructure safety
3	The minimum vertical separation between potable and non-potable pipelines at crossings is 18 inches.	Infrastructure safety /Public health
4	The water line shall be located above the wastewater line.	Infrastructure safety /Public health
5	All crossings under the high-speed (exceeding 50 mph) traffic lane must be encased.	Infrastructure safety /Public safety
6	The minimum horizontal separation between 6-inch to 10-inch water mains and sanitary sewer mains shall be 10 feet.	Public health/ Maintainability
7	Water service lines shall maintain a minimum horizontal separation of 6 feet from sanitary sewer laterals.	Public health/ Maintainability

Geospatial data of utilities and transportation and textual data of spatial constraints were processed using TripleGeo and TripleText to generate interlinked ontology instances in RDF. Figure 2.22 illustrates the graph view of the conversion result for the water main of WM4110, the driving lane of DL1, and the spatial constraint No.2.

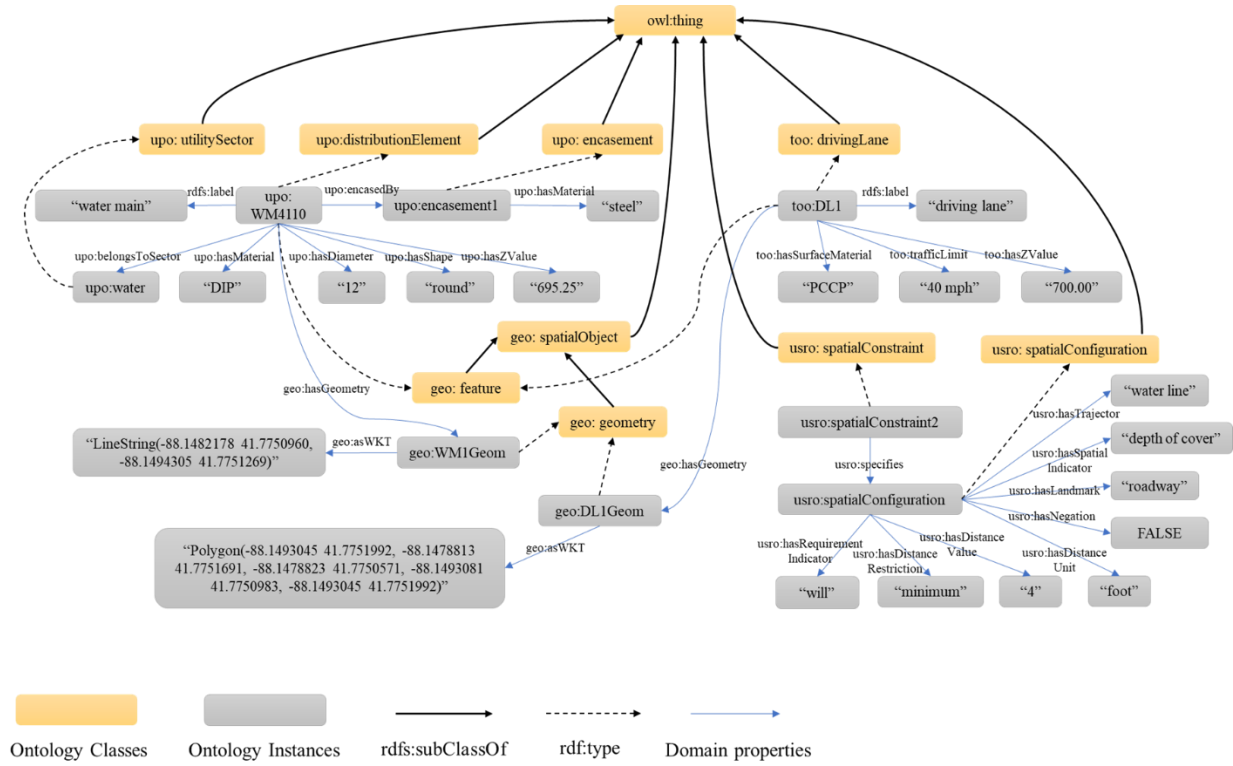


Figure 2.22. A partial graph view of the conversion result

Resulting RDF data were loaded into the Apache Jena platform for compliance checking. Spatial constraint RDF data was first analyzed and converted into SPARQL queries. The generation process is based on the description in section 2.6. Table 2.3 gives the generated SPARQL queries for this case study.

Table 2.3. Generated SPARQL queries for compliance checking

No.	Checked objects	SPARQL queries
1	Manhole	<pre> SELECT ?manhole WHERE { ?manhole a upo:utilityProduct; rdfs:label ?mappedTerm1. FILTER (?mappedTerm1 = "sewer manhole" ?mappedTerm1 = "storm manhole"). ?manhole geo:hasGeometry ?MHGeo. ?wheelPath a too:transportationObject; rdfs:label ?mappedTerm2. FILTER (?mappedTerm2 = "wheel path" ?mappedTerm2 = "traffic lane" ?mappedTerm2 = "driving lane"). ?wheelPth geo:hasGeometry ?WPGeo. ?intersection a too:transportationObject; </pre>

		<pre> rdfs:label ?mappedTerm3. FILTER (?mappedTerm3 = "intersection" ?mappedTerm3 = "crossroad"). ?intersection geo:hasGeometry ?INGeo. NOT EXISTS {?MHGeo geo:sfDisjoint ?WPGeo.} UNION NOT EXISTS {?MHGeo geo:sfDisjoint ?INGeo.} } </pre>
2	Water line	<pre> SELECT ?waterLine WHERE { ?waterLine a upo:utilityProduct; rdfs:label ?mappedTerm1. FILTER (?mappedTerm1 = "water line" ?mappedTerm1 = "water pipeline" ?mappedTerm1 = "water main" ?mappedTerm1 = "water lateral line" ?mappedTerm1 = "water service line"). ?roadway a too:transportationObject; rdfs:label ?mappedTerm2. FILTER (?mappedTerm2 = "roadway" ?mappedTerm2 = "travelled way" ?mappedTerm2 = "road"). FILTER (spin:depthOfCover(?waterline, ?roadway, unit:foot) < 4). } </pre>
3	Potable and non-potable pipelines	<pre> SELECT ?waterLine ?sewerLine WHERE { ?waterLine a upo:utilityProduct; rdfs:label ?mappedTerm1. FILTER (?mappedTerm1 = "water line" ?mappedTerm1 = "water pipeline" ?mappedTerm1 = "water main" ?mappedTerm1 = "water lateral line" ?mappedTerm1 = "water service line"). ?waterLine geo:hasGeometry ?WLGeom. ?sewerLine a upo:utilityProduct; rdfs:label ?mappedTerm2. FILTER (?mappedTerm1 = "sewer line" ?mappedTerm1 = "sewer pipeline" ?mappedTerm1 = "sewer main" ?mappedTerm1 = "sewer lateral line" ?mappedTerm1 = "sewer service line"). ?sewerLine geo:hasGeometry ?SLGeom. ?WLGeom geo:sfIntersects ?SLGeom. FILTER (spin:verticalDistance(?waterLine, ?SewerLine, unit:inch) < 18). } </pre>
4	Water line and wastewater line	<pre> SELECT ?waterLine ?sewerLine WHERE { ?waterLine a upo:utilityProduct; rdfs:label ?mappedTerm1. FILTER (?mappedTerm1 = "water line" ?mappedTerm1 = "water pipeline" ?mappedTerm1 = "water main" ?mappedTerm1 = "water lateral line" ?mappedTerm1 = "water service line"). ?sewerLine a upo:utilityProduct; rdfs:label ?mappedTerm2. FILTER (?mappedTerm1 = "sewer line" ?mappedTerm1 = "sewer pipeline" ?mappedTerm1 = "sewer main" </pre>

		<pre> ?mappedTerm1 = "sewer lateral line" ?mappedTerm1 = "sewer service line"). NOT EXISTS {?waterLine spin:above ?sewerLine.} } </pre>
5	Crossed utility pipes	<pre> SELECT ?crossedLine WHERE { ?crossedLine a upo:distributionElement; geo:hasGeometry ?CLGeom. ?trafficLane a too:transportationObject; rdfs:label ?mappedTerm. FILTER (?mappedTerm = "wheel path" ?mappedTerm = "traffic lane" ?mappedTerm = "driving lane"). ?trafficLane geo:hasGeometry ?TLGeom; too:hasTrafficLimit ?trafficLimit. FILTER (?trafficLimit > 50). ?CLGeom geo:sfIntersects ?TLGeom. ?crossedLine spin:below ?trafficLane. NOT EXISTS {?crossedLine upo:encasedBy ?encasement.} } </pre>
6	Water main and sanitary sewer main	<pre> SELECT ?waterMain ?sewerMain WHERE { ?waterMain a upo:utilityProduct; rdfs:label "water main"; upo:hasDiameter ?WMDia; geo:hasGeometry ?WMGeom. ?sewerLine a upo:utilityProduct; rdfs:label "sewer main"; upo:hasDiameter ?SMDia; geo:hasGeometry ?SMGeom. FILTER (?WMDia >6 && ?WMDia < 10 && ?SMDia >6 && ?SMDia < 10 && geof:distance(?WMGeom, ?SMGeom, unit:feet) < 10). } </pre>
7	Water service line and sanitary sewer lateral	<pre> SELECT ?waterLateral ?sewerLateral WHERE { ?waterLateral a upo:utilityProduct; rdfs:label ?mapperTerm1; geo:hasGeometry ?WLGeom. FILTER (?mappedTerm1 = "water lateral line" ?mappedTerm1 = "water service line"). ?sewerLateral a upo:utilityProduct; rdfs:label ?mapperTerm2; geo:hasGeometry ?SLGeom. FILTER (?mappedTerm2 = "sewer lateral line" ?mappedTerm2 = "sewer service line"). FILTER (geof:distance(?WLGeom, ?SLGeom, unit:feet) < 6). } </pre>

The generated SPARQL queries were executed against utility and transportation data to detect and separate non-compliant cases from compliant cases. Besides checking spatial constraints based on locations, the proposed method also has the capacity to check spatial constraints embodied in attribute values. For example, under constraint No. 5, if a utility crossing is under the high-speed traffic lane, then it must meet the pipeline encasement requirement and

this information is captured in the attribute of “Encas.” In this case, 23 noncompliant utility products were detected through SPARQL query, as summarized in Table 4 (the third column). These products were then connected to their corresponding spatial constraints via the `isNonCompliantWith` relationship for reporting and future retrieval.

To validate the feasibility and effectiveness of the newly developed method, the comparison between the results of compliance checking using the developed method and manual operation in ArcGIS was conducted. The authors designed a series of spatial queries in ArcGIS to check the data in the AOI against the spatial constraints. For example, using the “Select By Location” tool in ArcGIS, all the manhole that violate spatial constraint No.1 can be identified by selecting features in the manhole layer that fall within the driving lane or intersection layer features. Through spatial query in ArcGIS, 23 noncompliant features were identified as summarized in Table 2.4 (the fourth column). Table 2.4 illustrates the two sets of results side-by-side, organized under the corresponding spatial constraints. It is clear that the results match. The newly developed method achieves 100% performance in compliance checking.

Table 2.4. Comparison results

No.	Checked object	Non-compliant ontology instances in the case study	Non-compliant features through manual judgement in ArcGIS	Comparison result
1	Manhole	SewerMH1674, SewerMH512, SewerMH1378, StormMH56804, StormMH57334, StormMH57335	Sewer MH-1674, Sewer MH-512, Sewer MH-1378, Storm MH-56804, Storm MH-57334, Storm MH-57335	Consistent
2	Water line	/	/	Consistent
3	Potable and non-potable pipelines	(WM4110, SLL99090), (WM4110, SLL76872), (WM4110, SLL79618), (WM4110, SLL79617), (WM4110, SLL96288), (WM4110, SM214-1674)	(Water M-4110, Sewer LL-99090), (Water M-4110, Sewer LL-76872), (Water M-4110, Sewer LL-79618), (Water M-4110, Sewer LL-79617), (Water M-4110, Sewer LL-96288), (Water M-4110, Sewer Main 214-1674)	Consistent
4	Water line and wastewater line	/	/	Consistent
5	Crossed line	WLL13788, WLL11735, SLL77625, SLL79824, SLL79845, StormM57335	Water LL-13788, Water LL-11735, Sewer LL-77625, Sewer LL-79824, Sewer LL-79845 Storm Main-57335	Consistent
6	Water main and sanitary sewer main	/	/	Consistent
7	Water service line and sanitary sewer lateral	(WLL11249, SLL76872), (WLL13725, SLL78543)	(Water LL-11249, Sewer LL-76872), (Water LL-13725, Sewer LL-78543)	Consistent

2.9 Discussion

The intellectual contribution of this study is threefold.

(1) The ontology-based data exchange mechanism addresses the issue of data heterogeneity in the utility domain by providing a unified semantic schema and data convertors for heterogeneous utility data. Compared to open standard-based exchange mechanism (e.g., CityGML/Utility Network ADE), the main benefits of ontology-based mechanism are described as follows. First, although XML schemas are sufficient for exchanging data between parties who must have agreed to the definitions beforehand, their lack of semantics prevents machines from reliably performing this task with new XML vocabularies. Ontology is able to provide a shared vocabulary among the parties by defining abstract concepts/relationships such as the taxonomy of concepts, equivalent/disjoint concepts, and enumerations of terminologies. As such, data is given

explicit meaning, making it easier for machines to automatically process and integrate data through ontology. Second, using open standards, data creators and data receivers must have a deep understanding about the data schema (e.g., XML syntax) in order to avoid ambiguity and semantic inconsistency during data exchange. While using ontology, the schema definition languages – RDFS/OWL and data model – RDF are easy-to-understand and RDF has features that facilitate data integration even if the underlying schemas differ [59]. This characteristic also makes the conversion from proprietary formats into RDF simple-to-develop. Last, ontology is usually expressed in a logic-based language. Semantic tools can perform automated reasoning using the ontology, and thus provide advanced services to intelligent applications such as the compliance checking of underground utilities.

(2) The SPARQL-based query mechanism with spatial extensions enables the semantic checking of spatial location-related data of utilities. Most of existing efforts of ontology-based compliance checking focused on querying or reasoning about non-spatial attributes of building data due to the limited capability of spatial reasoning when dealing with geometry data. With the spatial extensions, ontology-based compliance checking can be extended to more scenarios such as where the checking of building geometry data is required.

(3) The framework that integrates the interlinked ontologies, data convertors, and SPARQL spatial extensions fills in the research gap in the area of utility compliance checking, which provides a more transparent paradigm rather than the otherwise procedural/hard-coding implementation approach. Currently, most compliance checking environments follow a hard-coding implementation approach that involves more arbitrary programming work, which is unreachable for anyone but system programmers. In this study, the use of RDFS/OWL ontologies, RDF data model, and SPARQL facilitates a more transparent reasoning process that are easy-to-understand and simple-to-implement even by non-experts.

This paper mainly focuses on the checking of underground utilities against the spatial constraints in utility regulations. An NLP algorithm was used to extract linguistic elements from the sentences of spatial constraints to ease the subsequent RDF conversion. It was found that for sentences that contain more than two spatial configurations, errors such as missing landmarks and incorrect spatial indicators exist. Future research is needed to create a near 100% accurate and automated method for converting complex textual sentences into structured knowledge. In order to broaden the checking scope of underground utilities, future research is also needed to

incorporate/develop ontologies and data convertors for other utility disciplines such as construction and maintenance. As always argued by the research area of compliance checking, an effective mechanism that handles the mismatch between data terms used in domain engineering models and regulatory documents is still missing [11]. While a semantic resource serves an effective semantic reference, their vocabulary size is still limited. Future research is needed to automatically develop and maintain the semantic resource to keep up with the growth of new terms.

2.10 Summary and Conclusions

This paper presents an ontology-based approach to integrate heterogeneous geospatial data as well as textual data to enable automated compliance checking of underground utilities through logic and spatial reasoning. The following technical issues were addressed in this study: 1) four interlinked ontologies were developed to provide the semantic schema; 2) two data convertors – TripleGeo and TripleText – were customized to enable the conversion of heterogeneous data from proprietary formats into the common and interoperable format of RDF; and 3) a SPARQL-based query mechanism with spatial extensions was designed to detect utility spatial defects. An experiment on a sample utility database was also conducted to demonstrate the feasibility and effectiveness of the proposed approach in detecting utility spatial defects. The compliance checking results remain consistent with the results checked through manual judgement.

In terms of extendibility, the ontology models the most fundamental concepts in the domain in a flexible format to enable future evolution and extension of the ontology for representing other application-specific and/or enterprise-specific knowledge. For instance, the developed USRO can be extended to accommodate more complex utility regulations so that a broad range of utility compliance checking is reachable. In addition to compliance checking, the semantic framework can be extended to integrate heterogeneous data from multiple sources and support applications where spatial and logic reasoning are required. An example is the integration and reasoning of multi-mode sensing data (through the incorporation of the sensor ontology) in construction for various project management and control tasks.

3. TOWARDS A DOMAIN ONTOLOGY FOR UTILITY INFRASTRUCTURE: COUPLING THE SEMANTICS FROM CITYGML UTILITY NETWORK ADE AND DOMAIN GLOSSARIES

This chapter presents a novel method to develop a utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain. The novel method integrates a top-down strategy and NLP to develop the desired ontology from CityGML Utility Network ADE (a candidate open standard for modeling utility networks) and domain glossaries (lists of utility-specific terms and their textual definitions). First, a base ontology is formalized by abstracting the modeling information in the ADE through a series of semantic mappings. Second, a novel integrated NLP approach is devised to automatically learn the semantics from the glossaries. The learning process includes the extraction of utility product terms using conditional random field (CRF) and the classification of semantic relationships between the terms using long short-term memory (LSTM) networks. Finally, the semantics learned from the glossaries are incorporated into the base ontology to result in a domain ontology for utility infrastructure. The NLP approach was evaluated using human-annotated test sets, and results show an average accuracy of 96% in term extraction and 86% in semantic relationship classification. For case demonstration, a glossary of water terms was learned to enrich the base ontology (formalized from the ADE) and the resulting ontology was evaluated to be an accurate, sufficient, and shared conceptualization of the domain. The newly developed ontology functions effectively as an interoperability facilitator for the utility infrastructure domain, attributed to the semantic compatibility with existing utility modeling initiatives and enriched/expandable (using NLP) semantic vocabulary.

This work is under review in ASCE Journal of Computing in Civil Engineering, 2020, Xin Xu and Hubo Cai. *“Towards a domain ontology for utility infrastructure: coupling the semantics from CityGML Utility Network ADE and domain glossaries”*. Table titles and figure captions have been modified to maintain the form of the dissertation.

3.1 Introduction

Over the last few decades, the utility infrastructure domain has grown in the amount of computer software, technologies, and automation to help improve the management of massive utility infrastructure. Object-oriented digital models (3D, 4D, and nD) have been increasingly implemented across the design, construction, and operation and maintenance stages for a wide range of purposes such as visualization, clash detection, constructability review, and digital inspection [10,25,93]. Such digital model-based systems are of great benefit to individual stakeholders (e.g., public agencies, utility owners, contractors, and asset managers); however, due to the fragmented nature of the utility industry, current approaches to generating these digital models are mostly dependent on proprietary software programs, non-compatible program languages, or standards specific to a single domain [34,94,95]. Data sharing and exchange among the heterogeneous landscape of information modeling, known as interoperability, becomes a major challenge for a more integrated management in utility infrastructure [34,95]. Therefore, it is imperative to address the interoperability issue to allow for seamless transfer of heterogeneous data among various sources.

A promising approach to achieving interoperability is for each source to utilize a shared and reliable knowledge model, known as an ontology, which defines and standardizes domain knowledge to function as the semantic enabler of communication between different sources [27,96]. A few domain ontologies have been introduced for the utility domain [2,26,34–36]. However, they are very limited to facilitate data exchange in heterogeneous environments for the following two reasons. First, they are mainly implemented as a means for knowledge representation and neglect the compatibility with existing utility modeling initiatives [10]. Much laborious work is required to align the semantic schemas in the ontologies with the data schemas in various utility models for data exchange [26,35]. Second, their semantic vocabularies of domain terms and semantic relationships are relatively too small to interpret the meaning of data and avoid mismatches/no matches when integrating a multitude of data that have different terms [37]. There is a critical need in the utility domain for an ontology that can be utilized as the shared and reliable knowledge model to facilitate a high degree of interoperability.

To fulfill that demand as well as to overcome the limitations in existing ontologies, this paper develops an ontology for the utility domain by coupling the semantics of CityGML Utility Network ADE [68], which is a candidate open standard for modeling utilities, and domain

glossaries, which archive important utility terms and their textual definitions. First, a base ontology is formalized by abstracting the modeling information in the ADE through a series of semantic mappings. Second, a novel integrated natural language processing (NLP) approach is devised to automatically learn the semantics from the glossaries. The learning process includes the extraction of utility product terms using conditional random field (CRF) and the classification of semantic relationships between the terms using long short-term memory (LSTM) networks. Finally, the semantics learned from the glossaries are incorporated into the base ontology to result in a domain ontology for utility infrastructure. In developing the proposed method, the methods for concept extraction, matching, and classification proposed by Zhang and EI-Gohary [43] were used and adapted to develop the desired ontology for the utility infrastructure domain. Please see Section 3.2.2 for further details. The NLP approach was evaluated using human-annotated test sets. For case demonstration, a glossary of water terms was learned to enrich the base ontology (formalized from the ADE) and the resulting ontology was evaluated to be an accurate, sufficient, and shared conceptualization of the domain. Attributed to the semantic compatibility with existing utility modeling initiatives and enriched/expandable (using NLP) semantic vocabulary, the developed ontology functions effectively as an interoperability facilitator for the utility infrastructure domain.

3.2 Background and Related Studies

3.2.1 The interoperability and ontology in the utility infrastructure domain

Due to the fragmentation of the utility industry, different stakeholders (e.g., public agencies, utility owners, contractors, and asset managers) may use proprietary software platforms with different data models to manage their own data [34,95]. Interoperability has been shown to be a major barrier for the seamless exchange of data between isolated and proprietary sources. Research efforts have been made to standardize the modeling of utility networks across the industry. Example standard models include INSPIRE Utility Networks, IFC, ArcGIS Utility Networks, SEDRIS, PipelineML, MUDDI [97]. However, these models tend to focus on a specific network type (water, electric, gas, communication, etc.), and/or a specific geographic scale (building, city, country, etc.), and thus, a complete and unified model for heterogeneous utility networks has not been formulated [97]. The CityGML Utility Network ADE extends the CityGML model to provide the required concepts for modeling different types of utility networks (such as electricity, water,

wastewater, gas or telecommunication) via 3D city models [68]. While the ADE offers a potential common basis for the integration of the diverse models, such an integration remains at the syntactic level – heterogeneous data are structured as loosely coupled documents that are not semantically compatible as different sources may use their unique sets of vocabularies. Based on a review of existing alternatives for utility network modeling, Becker et al. [68] concluded that a suitable common/shared model does not yet exist to facilitate the interoperability among the heterogeneous utility network models.

Recently, ontology has emerged as a promising tool to achieve semantic interoperability over fragmented, heterogeneous environments. An ontology describes the concepts, relationships, data properties and restrictions within a domain in a machine-readable manner [25,26], which can be utilized as the shared data format for each source to integrate data in heterogeneous formats. An increasing number of information management/exchange applications in construction have been relying on ontologies to support data interoperability, flexible data exchange, distributed data management, and the development of reusable tools [11,27,28]. In the GIS community, ontology has also been exploited to integrate a large amount of heterogeneous geospatial data. The efforts by this community have led to the development of CityGML ontology [29–31]. For instance, using CityGML ontology as the central platform, Métral et al. [30] integrated urban infrastructure data, transportation data, and urban planning data for creating a semantically enriched 3D city model.

A few ontologies have also been introduced for the utility domain [2,26,34–36]. But they are not suitable as domain-wide interoperability facilitators for the following two reasons. First, they lack semantical compatibility with existing utility modeling initiatives. For instance, some of the ontologies [2,34] mainly target at the knowledge representation purpose and neglect the alignment with existing utility data models. Additional alignments with various utility data models are required in order for the proposed ontologies to be the common/shared knowledge models for data exchange [26,35]. Second, their vocabulary sizes are too limited, which may lead to many mismatches or no-matches when integrating data from disparate sources that use different sets of vocabularies [37]. Mounce et al. [36] presented an approach for ontology enrichment (in aspects of domain terms and semantic relations) from domain corpora; however, their implementation still remains at the conceptual level without providing a practical solution to expand the ontology vocabularies. Therefore, in order to facilitate a high degree of interoperability across the utility

domain, the desired ontology must maintain semantic compatibility with existing utility modeling initiatives as well as a sufficient (or expandable) vocabulary size.

3.2.2 Natural language processing in ontology development

Several ontology development methodologies have been suggested [38,39]. They all include five key steps: (1) purpose and scope definition, (2) taxonomy building, (3) relation modeling, (4) ontology coding, and (5) ontology evaluation. Following these steps, EI-Diraby et al. [98] developed a formal taxonomy/ontology for construction knowledge as part of the e-COGNOS project. On top of that, extended work has been undertaken to develop ontologies for knowledge management in highway construction [99], processes in infrastructure and construction [86], and construction concepts in urban infrastructure products [34]. The five-step method for ontology development requires significant manual efforts on knowledge retrieval and ontology construction and validation. In attempts to reduce laborious work on ontology development, researchers have sought to design natural language processing (NLP) algorithms to build ontologies from a corpus of natural language text. NLP deploys artificial intelligence to enable computers to understand, create, and analyze human languages [40]. In the architecture, engineering, and construction (AEC) domain, a number of research studies have implemented NLP for document classification [100,101], information retrieval [102–104], and information extraction [49,50,105].

NLP contributes to ontology development in automated extraction of ontology contents – concepts and relations from textual documents and thus reduces laborious work on extracting ontology contents from textual documents. Concept extraction is a well-established field in computational linguistics, which can be implemented over fully unstructured documents [37,43,105,106] or semi-structured/structured documents such as glossaries [107] and table of contents [108]. Traditional approaches often follow a two-step procedure to extract concepts: first to extract technical terms from textual documents utilizing syntactic patterns, and then to identify the important concepts based on TF-IDF, C-Value, or Termex [37]. By contrast, relation extraction is identified as a more difficult problem, especially for non-taxonomic relations [109]. Two main approaches exist for relation extraction: distributional approach and path-based approach. The distributional approach leverages the contexts of each concept/term – distributional representations of word semantics to determine the semantic relatedness between concepts [110] while the path-

based approach considers lexico-syntactic patterns between the joint occurrences of concept/term pairs for relation detection [111,112].

In the AEC domain, Abuzir and Abuzir [42] developed the ThesWB system which utilizes hand-coded syntax patterns to detect lexical relations between civil engineering terms from HTML web pages. Rezgui [113] suggested a more sophisticated approach which relies on TF-IDF to identify important concepts for the domain of interest and computes the relatedness between the concepts using metric clusters. Le and Jeong [37] proposed an integrated method that implements rule-based NLP to detect domain terms from textual documents and uses machine learning (ML) to determine the semantic relatedness among terms using their occurrence statistics in a corpus. Since it is challenging to directly build an ontology from the extracted concepts and relations (higher textual analysis and more human work are required) [41], most studies end up building plain (or unstructured) dictionaries that simply archive the extracted ontology contents [37,42]. A few studies have adopted a top-down strategy to build ontology from the extracted concepts and relations [43,44]. Existing semantic models (taxonomies/ontologies) are first selected as bases, and enrichment follows by using the contents extracted from textual documents. For instance, Zhang and EI-Gohary [43] utilized rule-based NLP to extract concepts/relationships from regulatory documents and extended the existing IFC taxonomy with the extracted contents. The top-down strategy can save significant time and effort in building the knowledge skeletons of the ontology – the ontology directly inherited the semantics (formal definitions of classes and relations) provided by the existing semantic models. This study used the methods for concept extraction, matching, and classification proposed by Zhang and EI-Gohary [43] and adapted them for developing the desired ontology for the utility infrastructure domain. The following presents the main differences between the proposed method and the methods by Zhang and EI-Gohary [43]:

- Addressing a different domain – utility infrastructure domain. The starting semantic model for ontology development/enrichment is CityGML Utility Network ADE, and the enrichment source texts are semi-structured textual documents – domain glossaries of utility terms.
- Designing a new NLP for concept/relation extraction from textual documents. The novel NLP consists of a CRF model for term extraction and LSTM networks for semantic relationship classification, which are implemented to automate the extraction of ontology contents from domain glossaries.

- Resulting in an enriched utility ontology. A complete workflow for incorporating the semantics extracted from domain glossaries into the base ontology developed from the CityGML Utility Network ADE is presented. This study results in the desired utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain.

3.3 Study Objectives and Contributions

This study aims to develop a utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain. A novel method that integrates the top-down strategy and NLP has been designed and tested to achieve the study objective.

Departing from the CityGML Utility Network ADE, a base ontology is first formalized, followed by the incorporation of the semantics that are extracted from domain glossaries using NLP. The ADE is selected due to its candidacy as the open standard for modeling utility networks. This would guarantee the semantic compatibility of the target ontology with the modeling initiatives of the utility industry. Domain glossaries are semi-structured documents that archive domain important terms as well as their textual definitions. This selection would reduce the chance of extracting irrelevant information, and to focus on the extraction of critical ontology contents from the textual definitions. A novel integrated NLP that consists of a CRF model for term extraction and LSTM networks for semantic relationship classification is implemented to automate the extraction of ontology contents from domain glossaries. By coupling the semantics of the ADE and domain glossaries, a domain ontology for utility infrastructure is formalized.

The contributions of this study include the following three aspects. First, the integration of the top-down strategy and NLP significantly reduces the laborious work during the process of ontology development. The approach can also be adapted to ontology development for other domains. Second, the integrated NLP approach enables fully automated extraction of ontology contents from domain glossaries, which can help maintain the ontology to keep up with the growth of new domain knowledge. The approach can also be customized/strengthened to extract meaningful information from other types of documents. Third, the ontology formalized by coupling the semantics of the ADE and domain glossaries is a superior interoperability facilitator

for the utility domain as compared to the existing ones. To be more specific, the ontology is semantically compatible with the modeling practice in the ADE and thus, it can serve as an effective intermedium for data exchange; and also, the ontology has an enriched semantic vocabulary (which can be expanded from domain glossaries in timely and automated manners), thus facilitating the semantic integration of data between disparate sources that use different sets of vocabularies.

3.4 Development of a Domain Ontology for Utility Infrastructure

Figure 3.1 presents the overall process towards the development of the utility ontology. It consists of three modules: base ontology development module, ontology learning module, and ontology enrichment module. The base ontology development module focuses on the semantic abstraction from the CityGML Utility Network ADE. In this module, a series of semantic mappings are utilized to re-structure the ADE in the format of base ontology. The ontology learning module implements NLP to extract the semantics from utility glossaries. It involves two main tasks: term extraction and semantic relationship classification. The outcome from this module includes a list of extracted terms and semantically classified term pairs. The ontology enrichment module aims to incorporate the extracted semantics from the glossaries into the base ontology to build the utility ontology.

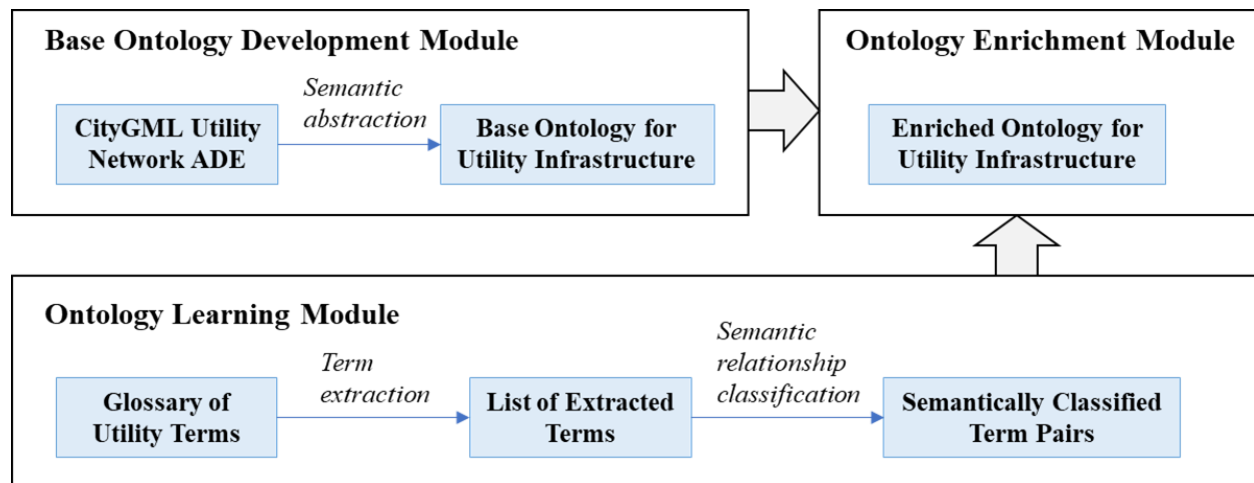


Figure 3.1. The development process of the utility ontology

3.4.1 Base ontology development

This section gives a brief introduction of the CityGML Utility Network ADE and its abstraction to the base ontology in OWL.

3.4.1.1 *CityGML Utility Network ADE*

The CityGML Utility Network ADE defines concepts which allow for modeling different types of utility networks in 3D city models [97]. The ADE is structured into six thematic modules as shown in Figure 3.2: (1) Network Core – defines the central concepts for representing utility networks, (2) Network Components – provides the individual components of utility networks, (3) Network Properties – defines the types of commodities transported by networks and their characteristics, (4) Feature Material – defines the exterior, interior and filling materials of network components, (5) Functional Characteristics – provides the functional concepts of supply area, functional roles, and suppliability/suppliedness of city objects, and (6) Geometry of Network Components. Figure 3.2 also presents the defined classes, attributes, and relations under Network Components in Unified Modeling Language (UML). In this module, the base class `AbstractNetworkFeature` is specialized into three classes `AbstractDistributionElement`, `AbstractFunctionalElement`, and `EnclosingElement`, all of which inherit the attributes from the base class. The class `AbstractFunctionalElement` includes two subclasses: `SimpleFunctionalComponent` – the superclass for simple functional components (e.g., `StorageComponent`) and `ComplexFunctionalComponent`. In order to specify the type of the functional components explicitly, the attribute class is defined under applicable classes (e.g., `ControllerComponent`). The code lists with common values of different types of functional components are provided for reference. The UML associations, aggregations, and compositions are also defined to present the detailed relations among the classes. The readers are referred to Becker et al. [68] for an in-depth introduction to the different ADE modules.

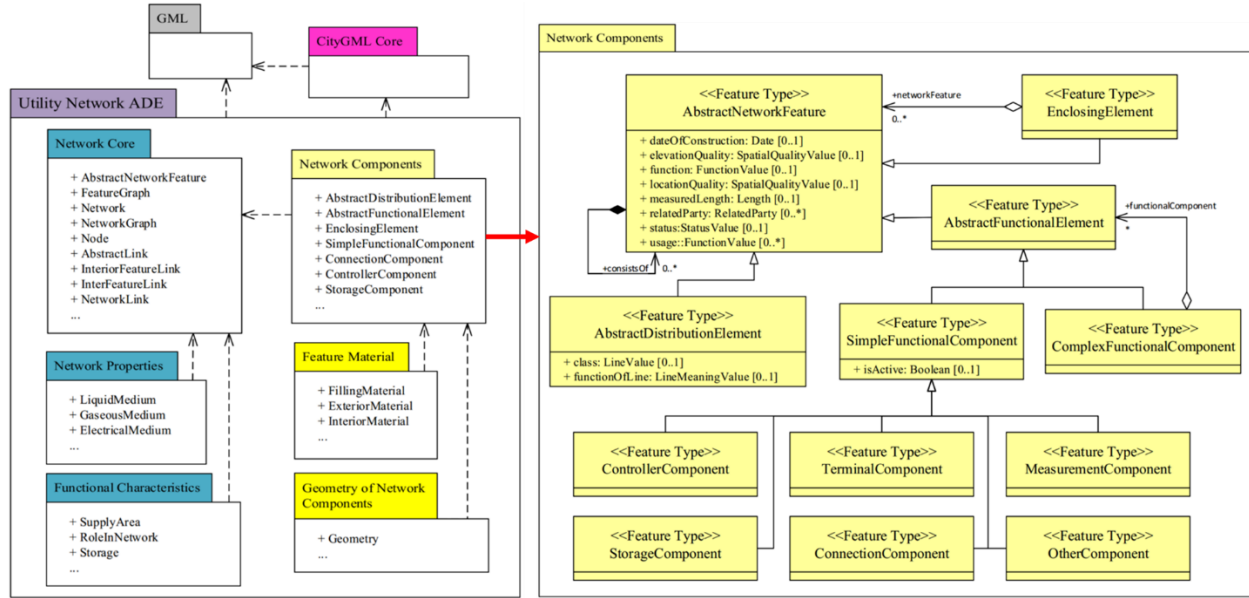


Figure 3.2. Modules of the CityGML Utility Network ADE and the UML diagram (partial) of Network Components [68]

3.4.1.2 Base ontology in OWL

The ADE has defined its data model in UML. The base ontology is developed by restructuring the ADE in OWL – a machine-readable language that supports the modeling of classes, attributes, and relationships in ontologies. For illustration, the following introduces the OWL formalization of the ADE Network Components module. Table 2.1 presents the UML-to-OWL mappings (partial) utilized to re-structure this module. The UML classes correspond to the OWL classes. The UML generalizations, aggregations/compositions, and associations correspond to OWL object properties (built-in/user-defined) that are declared between applicable OWL classes. The UML attributes correspond to OWL datatype properties, whose range can be predefined XML schema datatypes or user-defined datatypes based on the specific datatypes of the UML attributes. In addition, if the UML associations have specified cardinalities, they can be expressed as OWL restrictions on applicable OWL object properties. The OWL has more expressive elements to describe the semantics that have correspondences to the ADE and meanwhile, the OWL formalization has formal rigidity in logic theory that can support advanced application services.

Table 3.1. UML-to-OWL mappings (partial) for re-structuring the *Network Components* module




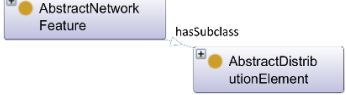
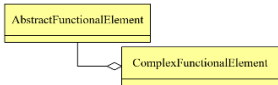
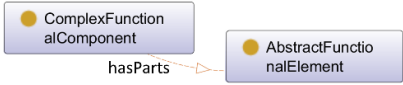
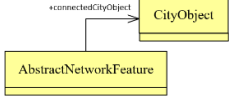
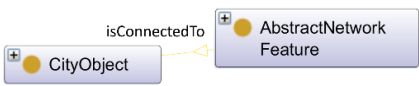
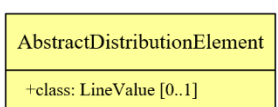
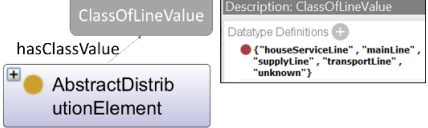
UML class diagram elements	OWL ontology elements
UML class 	OWL class 
UML generalization 	OWL object property (i.e., <i>hasSubclass</i>) 
UML aggregation/composition 	OWL object property (i.e., <i>hasParts</i>) 
UML association 	OWL object property (e.g., <i>isConnectedTo</i>) 
UML attribute and the datatype 	OWL data property (e.g., <i>hasClassValue</i>) and datatype (e.g., <i>ClassOfLineValue</i>) 

Figure 3.3 presents the base ontology (that is semantically equivalent to the Network Components module) in graphs (partial). Following the same process, the other AED modules can also be re-structured in OWL. The connections among the different modules can be built by declaring semantic relationships between cross-module concepts, thus resulting in the base ontology of the ADE.

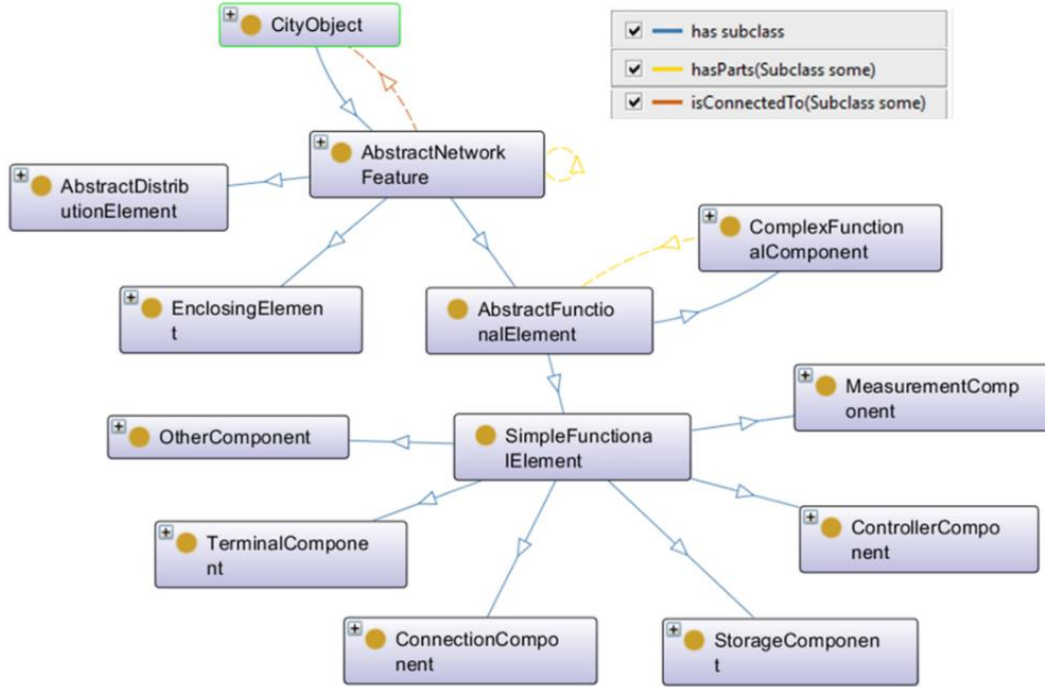


Figure 3.3. The base ontology (Network Components module) in graphs (partial)

3.4.2 Ontology learning

This section presents the ontology learning process from utility glossaries. Two consecutive tasks are involved: extraction of utility terms and semantic relationship classification of the terms. The learned semantics would serve to enrich the semantic vocabulary of the base ontology.

3.4.2.1 Term extraction

A utility glossary contains an alphabetical list of utility terms with their textual definitions, as structured in Figure 3.4. The glossary terms listed alphabetically (called key terms thereafter) can be extracted without effort while those mentioned in textual definitions (called mentioned terms thereafter) would be more difficult to handle. This task focuses on the extraction of mentioned terms relating to utility physical products (such as water pipes, electric cables, or sewer manholes) from textual definitions.

Usually, a utility product term may consist of one or a sequence of words and thus, extraction of terms can be implemented as a sequential labeling task, in which each word along

the input sequence (e.g., sentence) is assigned with a label indicating whether the word begins (B), is inside (I), or is outside (O) of a term. Figure 4 also gives a sample labeling of a sequence of words using the B, I, and O labels. The sequence contains three utility product terms, control valves, iron-bodied gate valves, and water mains.

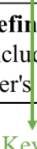
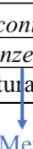

Term: <u>Control valve</u>												
Textual definition: In the water industry, <i>control valves</i> are generally <i>iron-bodied gate valves</i> installed in the <i>water mains</i> , but also include many types of smaller <i>bronze valves</i> that perform special functions in connecting and controlling water in the end user's <i>service line</i> . For use with natural gas, <i>control valves</i> are made of forged steel, cast or ductile iron, or brass.												
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>Key term</p> </div> <div style="text-align: center;">  <p>Mentioned term</p> </div> <div style="text-align: center;">  <p>Term extraction as sequential labeling</p> </div> </div>												
Word sequence: ... control valves are generally iron-bodied gate valves installed in the water mains ...												
Label sequence: ... B I O O B I I O O O B I ...												

Figure 3.4. Term extraction from the textual definitions

CRF is a class of discriminative probabilistic model best suited to sequential labeling tasks. CRF can be represented as an undirected graph, conditioned on a set of observations \mathbf{x} to predict a set of output labels \mathbf{y} . The simplest graph structure – linear-chain CRF, which was used in this study, is shown in Figure 3.5. A linear-chain CRF defines a conditional probability for a label sequence \mathbf{y} given an observation sequence \mathbf{x} to be:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{t=1}^N \sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}, t) \right) \quad (1)$$

where $Z(\mathbf{x})$ is the normalization factor that makes the probability of all label sequences sum to one; t ranges over the input sequence; $f_k(y_{t-1}, y_t, \mathbf{x}, t)$ is a feature function that measures any aspect of a label transition, $y_{t-1} \rightarrow y_t$, and the observation sequence \mathbf{x} , centered at position t ; λ_k is a learned weight associated with feature f_k . The parameters/weights can be estimated by maximum likelihood – maximizing the conditional probability $p(\mathbf{y}|\mathbf{x})$. In sequential labeling, the objective is to infer the most probable label sequence \mathbf{y}^* given an observation sequence \mathbf{x} , which can be determined by the following maximization:

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmax}} p(\mathbf{y}|\mathbf{x}) \quad (2)$$

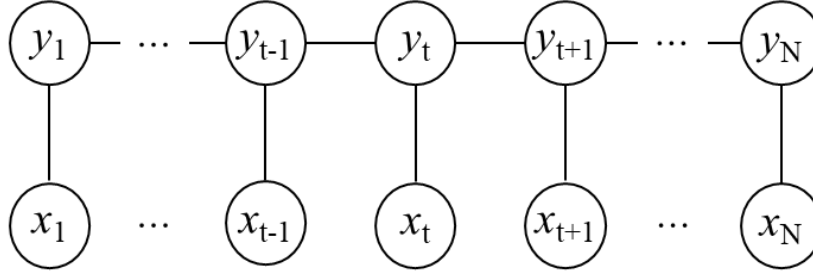


Figure 3.5. Linear-chain CRF graph structure

In the setting of term extraction, a sentence is a sequence of observations (i.e., words) \mathbf{x} , which can be described using the linear-chain CRF model. Each word can be represented as an input feature vector, and the CRF model outputs a label (e.g., B, I, or O) y for each word in the sequence. To define the feature vector for each word, the following syntactic features are used: original words, stems, part-of-speech (POS) tags, and lower/title/upper/digit/alnum flags, all of which can be extracted using off-the-shelf NLP tools. A context window of size one is constructed to include the features of the current word, as well as the features of the preceding and succeeding words for representing each word in the sentence. Figure 3.6 shows the feature representation for each word. The intent is to provide information on how the current word should be interpreted and labeled based on the features of the neighboring words, not only those of the current word. Along the sentence, words with their represented feature vectors and actual labels are fed into the CRF model for training. The L-BFGS method is used to calculate the optimal set of model parameters/weights from the training data because of its faster convergence to the global maximum. The Viterbi algorithm is used to obtain the most probable label sequence (i.e., $\langle B, I, I, \dots, I \rangle$) through Eq. (2) for extracting utility product terms.

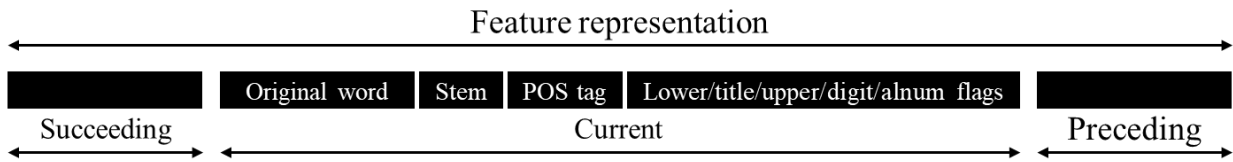


Figure 3.6. Feature representation for each word

3.4.2.2 Semantic relationship classification

Usually, the key terms in the glossary have their semantically related terms described in the textual definitions – the mentioned terms. In the previous task, mentioned terms are extracted

using CRF. This task aims to classify the specific semantic relationships of the key term-mentioned term pairs into predefined categories based on the textual definitions.

(1) Inventory of domain-specific semantic relationships

An inventory of domain-specific semantic relationships is defined for the key term-mentioned term pairs. Table 3.2 lists the specific semantic relationships, which includes four directed relationships (i.e., Hypernym-Hyponymy, Component-Whole, Content-Container, and Entity-Location) and two undirected relationships (i.e., Synonymy and Connection), as well as their corresponding descriptions and illustrative examples. The term pair's direction matters in a directed relationship, that is, $R(T1, T2)$ differs from $R(T2, T1)$ for a same directed relationship R and two different terms $T1, T2$. In addition, a seventh relationship – Other is included to stand for any relationship other than those presented in Table 3.2. Considering the directionality, a total of eleven semantic relationships – Hypernym-Hyponymy ($T1, T2$), Hypernym-Hyponymy ($T2, T1$), Component-Whole ($T1, T2$), Component-Whole ($T2, T1$), Content-Container ($T1, T2$), Content-Container ($T2, T1$), Entity-Location ($T1, T2$), Entity-Location ($T2, T1$), Synonymy ($T1, T2$), Connection ($T1, T2$), and Other ($T1, T2$) are used for classification.

Table 3.2. The specific semantic relationships, descriptions, and illustrative examples

Semantic relationships	Descriptions	Illustrative examples
Directed	Hypernym-Hyponymy	The semantic relationship between a generic term (Hypernym) and a specific instance of it (Hyponymy). Hypernym-Hyponymy (<i>system valve, gate</i>): Types of <i>system valve</i> include <i>gate</i> , <i>plug</i> , <i>ball</i> , <i>cone</i> , and <i>butterfly</i> .
	Component-Whole	The semantic relationship between a component and a larger whole. Component-Whole (<i>main valve, gate-type hydrant</i>): A <i>gate-type hydrant</i> is a hydrant having one <i>main valve</i> .
	Content-Container	The semantic relationship between an object (Content) and its physically stored area of space, the container. Content-Container (<i>water meter, meter box</i>): A <i>meter box</i> is the housing or container that encloses a <i>water meter</i> .
	Entity-Location	The semantic relationship between an object (Entity) and its located/placed/installed area, the location. Entity-Location (<i>venturi meter, pipe</i>): A <i>venturi meter</i> is a flow measuring device placed in a <i>pipe</i> .
Undirected	Synonymy	The semantic relationship between two terms that share the same meaning. Synonymy (<i>stop box, curb stop</i>): A <i>stop box</i> is also referred to as a <i>curb stop</i> .
	Connection	The semantic relationship between two physically connected objects. Connection (<i>suction pipe, wet well</i>): The <i>suction pipe</i> of a pump may be connected to the <i>wet well</i> .

(2) Feature selection – linguistic information along the shortest dependency path

Dependency parsing is the task of extracting a dependency parse of a sentence that represents its grammatical structure and defines the relationships between “head” words and “dependent” words [114]. As such, it has been frequently used to dissect sentence and to identify the semantic relationship between two words/terms. Figure 3.7 presents the dependency parsing result of an example sentence. Dependency relations among the words are illustrated using directed, labeled arcs from heads to dependents (such as *gate valve* $\xleftarrow{nsbjpass}$ *installed*, which means *gate valve* is the nominal subject of the passive verb *installed*). To determine the target term pair’s semantic relationship, it is mostly sufficient to use only the linguistic information along the shortest dependency path (SDP) of the term pair. For example, in Figure 3.7, the SDP between the terms *gate valve* and *mains*, represented by the red arrows, condenses most relevant information about the target relationship while diminishing less relevant noise such as the words *is* and *typically* and

the dependency relations auxpass and advmod. Moreover, the SDP is effective in capturing word semantic information close in context but far in sentence distance. Take the following long sentence as an example. For the term pair control valves and bronze valves in the sentence “...control valves are generally iron-bodied gate valves installed in the water mains, but also include many types of smaller bronze valves...”, their SDP, control valves \xleftarrow{nsbj} are \xrightarrow{conj} include \xrightarrow{dobj} types \xrightarrow{prep} of \xrightarrow{probj} bronze valves, is shortened in length and is also capable of capturing the long distance dependency relation between them. In this study, in order to determine the semantic relationships between key terms and mentioned terms, the term pairs’ SDPs are first extracted from their occurred definition sentences and three types of linguistic information along the SDPs including words themselves, POS tags, and dependency relations are then used as the indicative features.

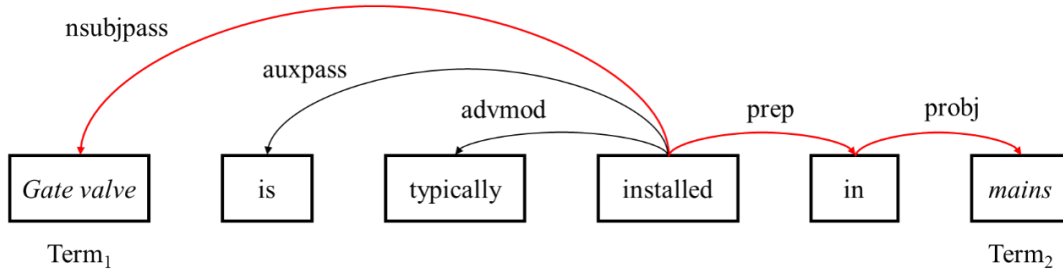


Figure 3.7. The dependency parsing result of an example sentence

(3) Learning architecture for classification – long short-term memory networks

LSTM networks are used to pick up heterogeneous linguistic information along the SDPs for semantic relationship classification. LSTM networks are special Recurrent Neural Networks (RNNs) with LSTM units that are capable of not only capturing long-term dependencies in sequential data but also addressing the gradient vanishing or exploding problem in classical RNNs. Figure 3.8 (a) presents the overall learning architecture using LSTM networks.

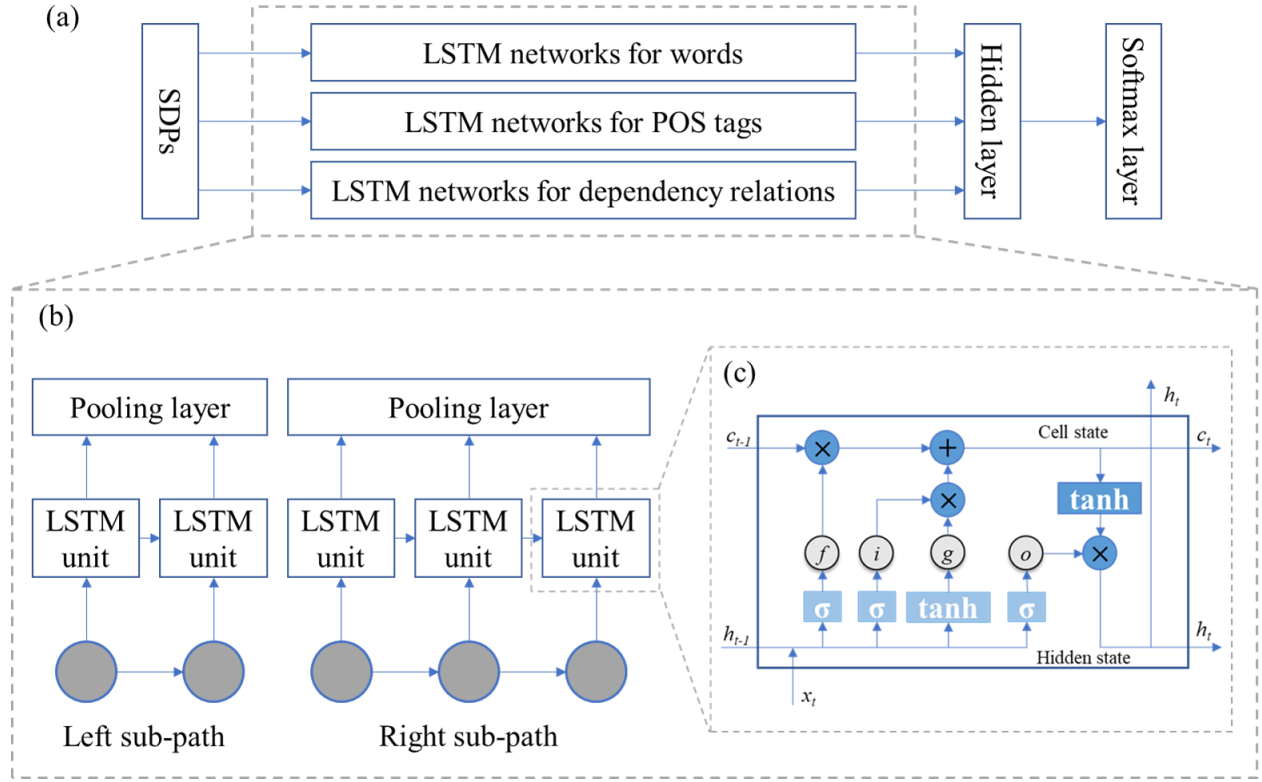


Figure 3.8. (a) The overall architecture for semantic relationship classification, (b) the LSTM networks for feature learning along the SDPs, and (c) the structure of an LSTM unit

The SDPs of the target term pairs serve as the inputs of the LSTM networks. Along the SDPs, linguistic features – words, POS tags, and dependency relations, are learned via their respective LSTM networks. The detailed process of feature learning via LSTM networks is depicted in Figure 3.8 (b). It is observed that an SDP can be separated into two sub-paths (left and right), each from the common ancestor “head” word to a target term, which provide strong hints for determining the directionality of the target semantic relationship. Considering this, two LSTM networks are designed to pick up information along the left and right sub-paths of the SDP, respectively, thus enabling the feature learning in a direction-sensitive manner. Take Figure 3.7 as an example. The target terms *gate valve* and *mains* have their common ancestor *head installed*, which separates the SDP into left sub-path installed $\xrightarrow{nsubjpass}$ *gate valve* and right sub-path installed \xrightarrow{prep} in \xrightarrow{probj} *mains*. Sequential features (such as word sequences: installed – *gate valve* and installed – in – *mains*) along the left and right sub-paths are learned separately. For effective

information propagation and integration along the sub-paths, LSTM units are leveraged. As depicted in Figure 3.8 (c), all LSTM units have the same structure that contains three gates – input gate, forget gate, and output gate – to control the flow and modify the information in and out of the unit. Along the sub-paths, sequential features, e.g., words, represented using real-valued vectors (called embeddings), are fed into their corresponding LSTM units. The LSTM units are updated through the following equations.

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \quad (4)$$

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \quad (5)$$

$$g_t = \tanh(W_g \cdot x_t + U_g \cdot h_{t-1} + b_g) \quad (6)$$

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1} \quad (7)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (8)$$

where h is the hidden unit; c is the memory cell; i is the input gate; f is the forget gate; o is the output gate; g is the candidate cell; σ denotes the sigmoid function; \otimes denotes element-wise multiplication. The three adaptive gates i_t , f_t , and o_t depend on the previous state h_{t-1} and the current input x_t . g_t is also calculated, serving as the candidate memory cell. The current memory cell c_t is updated based on the previous memory cell c_{t-1} and candidate memory cell g_t . The output of an LSTM unit is the hidden state h_t . $W_i, W_f, W_o, W_g, U_i, U_f, U_o, U_g, b_i, b_f, b_o, b_g$ are the learnable parameters for each LSTM unit that control the level of information transferred from previous states as well as the level of information taken from the current state. A max pooling layer thereafter gathers information from LSTM units along each sub-path. Since the three types of linguistic features along the SDP do not interact with each other during recurrent propagation, their respective pooling layers are concatenated, and then connected to a fully connected hidden layer. Finally, a softmax layer takes the output of the hidden layer as input and computes the probability of being any particular class of semantic relationship, e.g., Hypernym-Hyponymy (T1, T2).

3.4.3 Ontology enrichment

Figure 3.9 illustrates the overall process of ontology enrichment using the learned semantics from glossaries. It includes term incorporation (including key terms and mentioned

terms), followed by semantic refinement towards the resulting ontology, which is detailed as follows.

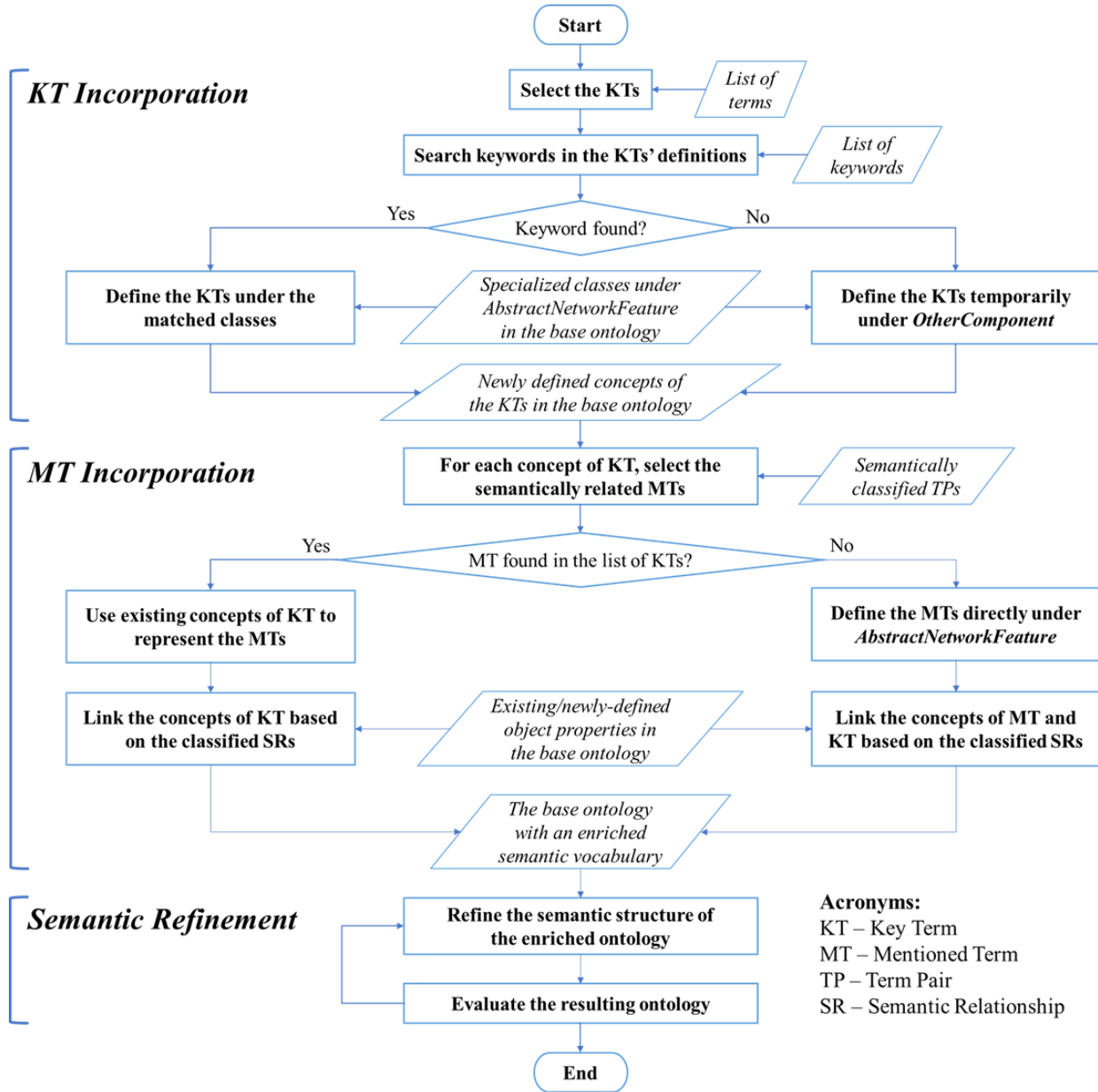


Figure 3.9. Ontology enrichment process

3.4.3.1 Incorporation of key terms

The key terms are first selected and defined as new concepts under *AbstractNetworkFeature* – the dedicated class in the base ontology for representing the physical components of utility networks. There are more specialized classes (e.g.,

AbstractDistributionElement, ControllerComponent) under AbstractNetworkFeature to categorize utility physical components based on their functions/usages in the utility networks (see Figure 3.3). For example, ControllerComponent represents those components (e.g., valves) used to control, limit or influence the flow of the transported commodity. The key terms should be defined under appropriate classes to retain the predefined taxonomic structure in the base ontology.

The textual definitions of the key terms contain keywords (mostly verbs) that indicate the functions of the defined utility products, and thus the keywords are used to determine under which specific classes the key terms are to be defined. For example, according to the definition of the term valve: a valve is a mechanical device installed in a pipeline to close off or regulate the flow of gas or liquid, the keywords “close off” and “regulate” indicate that valves function as flow controllers in the pipeline system and thus, valve should be defined under ControllerComponent. Table 3.3 presents a partial list of the function-indicative keywords for each specialized class under AbstractNetworkFeature.

Table 3.3. A partial list of keywords for each class under AbstractNetworkFeature

Classes in the base ontology	List of keywords
<i>AbstractDistributionElement</i>	distribute, carry, transport, conduct, convey
<i>EnclosingElement</i>	enclose, support, protect, wrap, encase, sleeve
<i>ConnectionComponent</i>	connect, join, couple, chain, interconnect, link, interlink
<i>ControllerComponent</i>	control, shutoff, close off, regulate, stop, divert, block, disconnect, break
<i>MeasurementComponent</i>	measure, sense, check, gauge, scale
<i>StorageComponent</i>	store, hold, stockpile, keep, stow, house
<i>TerminalComponent</i>	discharge, issue, emit, release, pour

As shown in Figure 3.9, for each key term that needs to be incorporated, keyword search is performed in the term’s textual definition. If any word in the keyword list (right column in Table 3.3) is found in the definition, the defined term would be added as a subclass under the matched class in the base ontology (left column in Table 3.3). Since the classes under AbstractNetworkFeature are not disjoint, a key term could be defined under multiple classes after keyword search. If no keyword is found, the term would be temporarily defined under OtherComponent.

3.4.3.2 Incorporation of mentioned terms

After the incorporation of key terms, the mentioned terms are incorporated into the base ontology following the process in Figure 3.9. For each concept of the incorporated key term, the semantically related mentioned terms are first selected and defined as new concepts of the base ontology. The concepts of mentioned terms are then linked to the concept of the key term using existing or newly defined object properties of the base ontology. The used object properties correspond to the classified semantic relationships of the key term-mentioned term pairs as the following: hasSubclass – Hypernym-Hyponymy, hasParts – Component-Whole, contains – Content-Container, hasLocation – Entity-Location, equivalentTo – Synonymy, isConnectedTo – Connection. For example, for the new concept Valve, which is defined from the key term valve, one of the semantically related mentioned terms is water valve, denoted as Hypernym-Hyponymy (valve, water valve); correspondingly, a new concept WaterValve is defined and a new semantic triple Valve–hasSubclass–WaterValve is constructed, thus enriching the base ontology with new semantics. Particularly, there exist some mentioned terms that are exactly the same as some key terms (they share the same lexical forms). Under this scenario, existing concepts of key terms are used to represent the mentioned terms and additional semantic links are built between the applicable concepts of key terms. In addition, some mentioned terms do not have specific semantic relationships with the key terms (the semantic relationship is classified as Other). The corresponding concepts thus do not have specific semantic links to other concepts.

3.4.3.3 Semantic refinement

New semantics – concepts and semantic relationships are added into the base ontology via term incorporation. The resulting semantic structure is not optimal and still needs further refinement as follows.

- Duplicate concepts. Some key terms may have a same set of mentioned terms, thus resulting in duplicate concepts of mentioned terms defined in the enriched ontology. Under this scenario, all duplicate concepts are made distinct and all semantic links are retained by the distinct concepts.
- Redundant semantic links. Especially there are redundant hasSubclass links in the enriched ontology. For example, three semantic triples Valve–hasSubclass–GateValve, Valve–

hasSubclass–ControlValve, and GateValve–hasSubclass–ControlValve are constructed through enrichment. The triple Valve–hasSubclass–ControlValve is redundant, which can be inferred from the other two triples. Under this scenario, all redundant semantic links are checked and removed.

- Empty assignments. The base ontology has defined ComplexFunctionalComponent to represent utility physical components that are composed of other functional components. Initially, no concepts are assigned under ComplexFunctionalComponent. After term incorporation, the concepts (especially those temporarily defined under OtherComponent) that are linked to other concepts via hasParts are also defined under ComplexFunctionalComponent. This refinement would achieve maximum compliance with the predefined semantic structure under AbstractNetworkFeature.

In the end, the refined ontology is evaluated through consistency check to ensure that 1) the ontology does not contain contradictory semantic triples and 2) the ontology is valid in OWL formalization. Figure 3.10 presents a partial graph view of the resulting ontology with incorporated semantics.

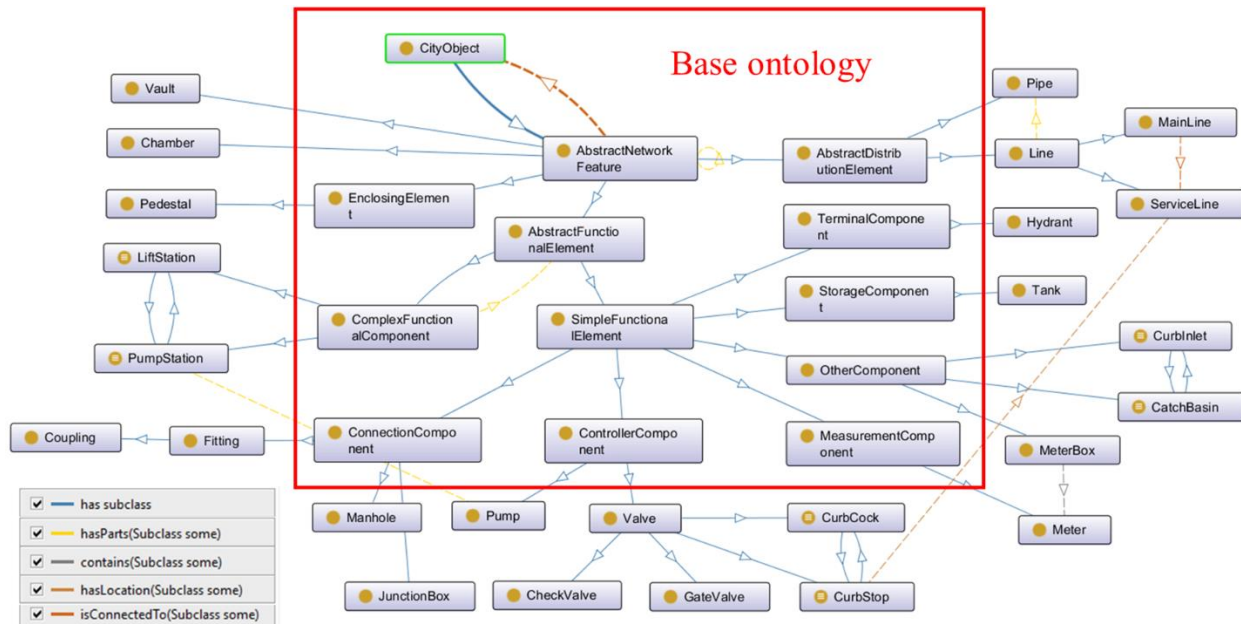


Figure 3.10. The resulting ontology with incorporated semantics (partially)

3.5 Experimentation and Case Demonstration

3.5.1 Term extraction

An experiment was conducted to evaluate the performance of the CRF-based approach in the extraction/labeling of mentioned utility product terms from the textual definitions in utility glossaries. In this experiment, a corpus was built by randomly collecting 200 definition sentences from utility glossaries. The corpus consists of nearly 8,100 tokens (including words and punctuation marks). Domain experts were invited to manually label the tokens using B, I, and O labels, which resulted in 912 utility product terms identified. A Python program Feature-Extractor was also developed to help extract the feature representations for individual tokens.

The corpus was split into two sets: (1) training set – 140 sentences and (2) test set – 60 sentences. By feeding the training set – a sequence of tokens' feature representations and labels into the CRF model, the optimal set of parameters/weights were learned. The test set was then used to evaluate the performance of the trained CRF model – Term-Labeler. Table 3.4 presents the evaluation results in terms of precision, recall, and F-measure. Let T_i denote a set of true tokens labeled with label i in the test set, and T_i' denote a set of tokens labeled with label i by the Term-Labeler. The precision (P_i), recall (R_i), and F-measure (F_i) for a certain label i are calculated using the following equations:

$$P_i = \frac{T_i \cap T_i'}{T_i'} \quad (9)$$

$$R_i = \frac{T_i \cap T_i'}{T_i} \quad (10)$$

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \quad (11)$$

The overall performance is evaluated based on the percentage of correctly labeled tokens for all types of labels relative to the total number of tokens existing in the test set. The test set contains a total of 2512 tokens, among which 241, 191, and 1987 tokens were correctly labeled as B, I, and O, respectively. As such, an average accuracy of 96.30% was achieved by the Term-Labeler.

Table 3.4. Evaluation results for term extraction

Label	Precision (%)	Recall (%)	F-measure (%)
B	0.870	0.880	0.875
I	0.946	0.868	0.905
O	0.977	0.985	0.981

Figure 3.11 presents the confusion matrix for the labeling results over the test set. Error analysis results in the following two findings. First, some terms may have adjectives as their constituent parts while some may not. This situation may lead to errors in labeling the adjectives (confusion between B and O). For example, the word sequence “potable water distribution pipe” (actual label sequence is <B, I, I, I>) could be sequentially labeled as <O, B, I, I> while the word sequence “smaller bronze valve” (actual label sequence is <O, B, I>) could be sequentially labeled as <B, I, I>. In order to reduce such errors, statistical features (e.g., TF-IDF) that measure the degree of a word/phrase being a domain-specific term can be incorporated into the training process for a more accurate term labeling. Second, the textual definitions also contain some terms relating to transportation products (such as pavements, shoulders, or curbs), which were incorrectly labeled as utility product terms. The Term-Labeler is unable to differentiate them merely based on their syntactic features. One possible way of improving the performance is to incorporate semantic features (that can be captured using ontologies) to enable the semantic labeling. Therefore, future research is still needed to further improve the accuracy and robustness of the Term-Labeler.

		<i>Predicted label</i>		
		B	I	O
<i>Actual label</i>	B	241	9	24
	I	7	191	22
	O	29	2	1987

Figure 3.11. Confusion matrix for term extraction in the test set

3.5.2 Semantic relationship classification

An experiment was conducted to evaluate the performance of the LSTM-based approach in classifying the specific semantic relationships of the key term-mentioned term pairs based on the textual definitions. In this experiment, a total of 1,000 definition sentences were collected from utility glossaries. For each definition sentence, the key term-mentioned term pair that needs to be semantically classified were marked and the true semantic relationship of the term pair was also assigned, thus forming the ground truth for this experiment. A Python program Path-Feature-Extractor was developed to help extract the SDPs of the marked term pairs as well as the linguistic features (words, POS tags, and dependency relations) along the left and right sub-paths of the SDPs from the definition sentences. For feature representations, words are mapped to 100-dimensional real-valued vectors that were pre-trained on the English Wikipedia corpus by Glove [115], and both POS tags and dependency relations are mapped to 25-dimensional real-valued vectors that were initialized randomly.

The collected definition sentences were split into two sets: (1) training set – 700 sentences and (2) test set – 300 sentences. The training set was first processed using the Path-Feature-Extractor and then fed into the LSTM networks for training. The training objective was to minimize the cross-entropy error; stochastic gradient descent was applied for optimization; and gradients were computed by standard back propagation. Once the Semantic-Relationship-Classifer was trained, it was evaluated using the test set. Table 3.5 presents the evaluation results in terms of precision, recall, and F-measure. The precision (P_i), recall (R_i), and F-measure (F_i) for a certain semantic relationship i are also calculated using Eq. (9–11), where T_i denotes a set of true term pairs classified with semantic relationship i in the test set, and T_i' denotes a set of term pairs classified with semantic relationship i by the classifier. The overall performance is evaluated based on the percentage of correctly classified term pairs for all types of semantic relationships relative to the total number of term pairs existing in the test set. The test set contains a total of 300 term pairs (one term pair per definition sentence), among which 258 pairs were correctly classified. As such, an average accuracy of 86% was achieved by the Semantic-Relationship-Classifier.

Table 3.5. Evaluation results for semantic relationship classification

Semantic relationship	Precision (%)	Recall (%)	F-measure (%)
Hypernym-Hyponymy (T ₁ , T ₂)	0.840	0.840	0.840
Hypernym-Hyponymy (T ₂ , T ₁)	0.875	0.903	0.889
Component-Whole (T ₁ , T ₂)	0.962	0.962	0.962
Component-Whole (T ₂ , T ₁)	0.782	0.900	0.837
Content-Container (T ₁ , T ₂)	0.833	0.556	0.667
Content-Container (T ₂ , T ₁)	0.839	0.897	0.867
Entity-Location (T ₁ , T ₂)	0.857	0.857	0.857
Entity-Location (T ₂ , T ₁)	0.778	0.583	0.667
Synonymy (T ₁ , T ₂)	0.975	0.929	0.951
Connection (T ₁ , T ₂)	0.839	0.897	0.867
Other (T ₁ , T ₂)	0.804	0.804	0.804

Figure 3.12 presents the confusion matrix for the classification results over the test set. Error analysis leads to the following findings. First, there are no confusion between the same relationship but with the term order inverted, such as Hypernym-Hyponymy (T₁, T₂) and Hypernym-Hyponymy (T₂, T₁). This demonstrates the effectiveness of separating the SDPs into two sub-paths in capturing the directionality of the relationships. Second, both the relationships Content-Container (T₁, T₂) and Entity-Location (T₂, T₁) show the lowest F-measure (66.7%). This is mainly because (1) their instances are very few, only accounting for around 4% of the total instances – their linguistic features were not sufficiently learned and generalized by the LSTM networks and (2) some instances were misclassified as other relationships (for example, Content-Container (T₁, T₂) was misclassified as Entity-Location (T₁, T₂)) – the decision boundaries between them were not clearly cut. Third, most confusion occurs between the ten explicitly defined relationships and the pseudo relationship Other (T₁, T₂). Other (T₁, T₂) stands for any relationship which is not one of the nine explicitly defined relationships. Adding it to the training set would force any model to correctly identify the decision boundaries between the explicitly defined relationships and “everything else”. This also encourages good generalization behavior to larger, noisier data sets commonly seen in real-world applications. However, the data for Other (T₁, T₂) prepared in this experiment is neither sufficient nor nonhomogeneous, thus leading to some confusion between the ten relationships and Other (T₁, T₂).

The Semantic-Relationship-Classifer can be improved in three aspects for an even higher efficacy: (1) extract more features such as domain semantics (using domain ontologies) and lexical semantics (using lexical databases such as WordNet) along the SDPs and incorporate them into

the LSTM-based learning process; (2) integrate non-SDP features such as the term pairs' distributional embeddings and surface string features (especially for those multiword terms that have common words) with the features along the SDPs and create an integrated method for feature learning; (3) increase the data size, improve the data quality (especially those for Other (T1, T2)), and provide a validation set to fine-tune the model hyperparameters during training, thus resulting in a more generalized model for relationship classification.

		<i>Predicted class</i>										
		HH1	HH2	CW1	CW2	CC1	CC2	EL1	EL2	SY	CO	OT
<i>Actual Class</i>	HH1	21	0	0	2	0	1	0	0	0	0	1
	HH2	0	28	0	0	0	0	0	0	1	0	2
	CW1	0	0	25	0	0	0	0	0	0	0	1
	CW2	1	0	0	18	0	1	0	0	0	0	0
	CC1	0	0	0	0	5	0	2	0	0	0	2
	CC2	0	0	0	2	0	26	0	1	0	0	0
	EL1	0	0	0	0	1	0	18	0	0	2	0
	EL2	0	0	0	0	0	2	0	7	0	0	3
	SY	0	3	0	0	0	0	0	0	39	0	0
	CO	0	0	0	0	0	0	2	0	0	26	2
	OT	3	1	1	1	0	1	0	1	0	3	45

HH1 = Hypernym-Hyponymy (T1, T2); HH2 = Hypernym-Hyponymy (T2, T1); CW1 = Component-Whole (T1, T2); CW2 = Component-Whole (T2, T1); CC1 = Content-Container (T1, T2); CC2 = Content-Container (T2, T1); EL1 = Entity-Location (T1, T2); EL2 = Entity-Location (T2, T1); SY = Synonymy (T1, T2); CO = Connection (T1, T2); OT = Other (T1, T2)

Figure 3.12. Confusion matrix for semantic relationship classification in the test set

3.5.3 Case demonstration

A case on the development of a domain ontology for utility infrastructure using CityGML Utility Network ADE and a glossary of water terms was demonstrated.

First, the base ontology was abstracted from the ADE using a series of UML-to-OWL mappings, thus resulting in 73 classes, 24 object properties, 43 data properties, and 21 datatypes. Most of the semantic declarations remain consistent with the ADE.

Then, additional semantics were learned from a glossary of water terms. The water dictionary [116] that was published by American Water Works Association (AWWA) was used, from which a total of 100 key terms (all related to utility physical products) as well as their textual definitions were randomly collected for learning. Term-Labeler was first used to extract/label the mentioned utility product terms from the textual definitions. As a result, 363 mentioned terms (excluding the key terms) were extracted and correspondingly, a total of 363 key term-mentioned term pairs were generated by pairing the key terms with their semantically related mentioned terms. Path-Feature-Extractor was then used to extract the learning features – SDPs of the term pairs from their co-occurred definition sentences. Finally, Semantic-Relationship-Classifer was used to determine the specific semantic relationships of the term pairs based on the extracted learning features. Figure 3.13 presents the learning results using an illustrative example. Among the 363 term pairs, 37, 115, 45, 23, 3, 24, 17, 0, 56, 14, and 29 instances were classified as Hypernym-Hyponymy (T1, T2), Hypernym-Hyponymy (T2, T1), Component-Whole (T1, T2), Component-Whole (T2, T1), Content-Container (T1, T2), Content-Container (T2, T1), Entity-Location (T1, T2), Entity-Location (T2, T1), Synonymy (T1, T2), Connection (T1, T2), and Other (T1, T2), respectively.

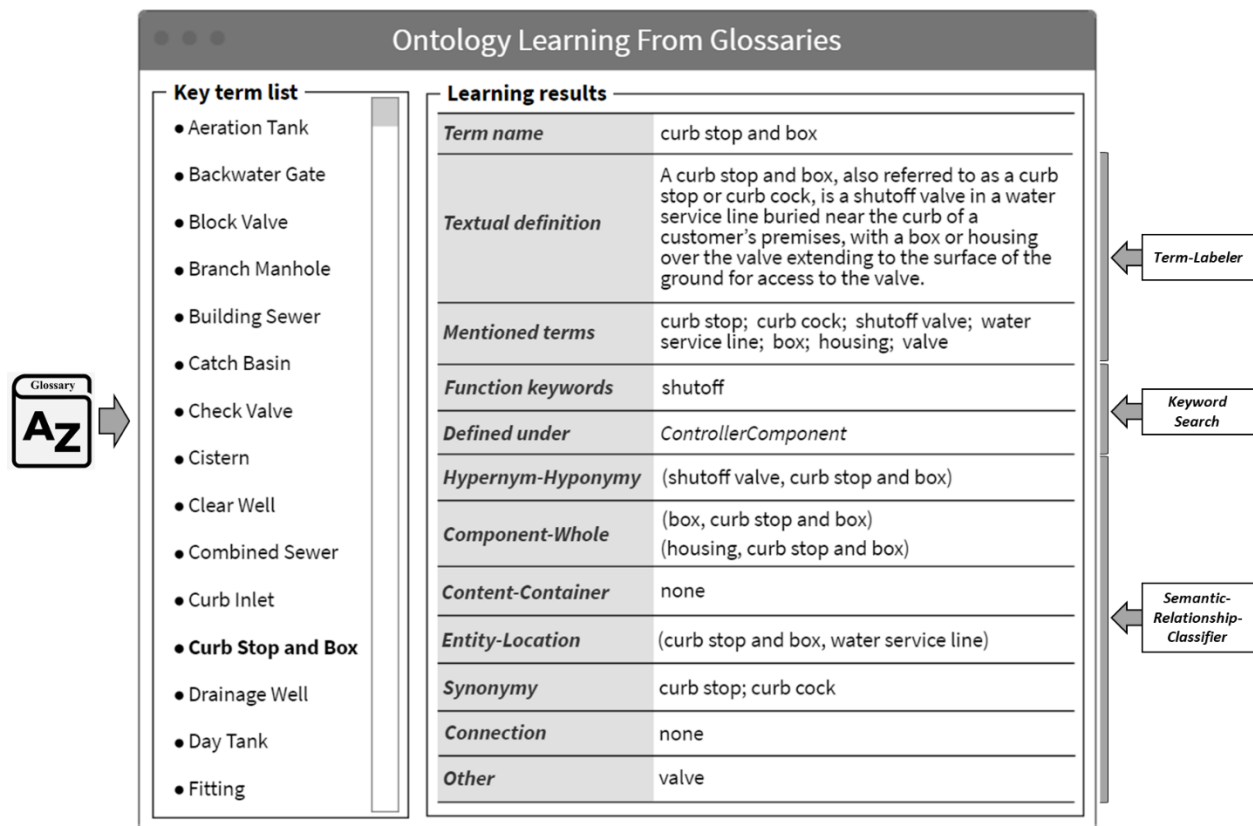


Figure 3.13. Ontology learning from glossaries

The last step was to incorporate the learned semantics into the base ontology. Following the process illustrated in Figure 3.9, a total of 428 terms (duplicates were made unique) and 2 semantic relationships (the others have their correspondences in the base ontology) were incorporated into the base ontology as new classes and object properties, respectively. Figure 3.14 presents the hierarchies of the classes, object properties, data properties, and datatypes in the resulting ontology.

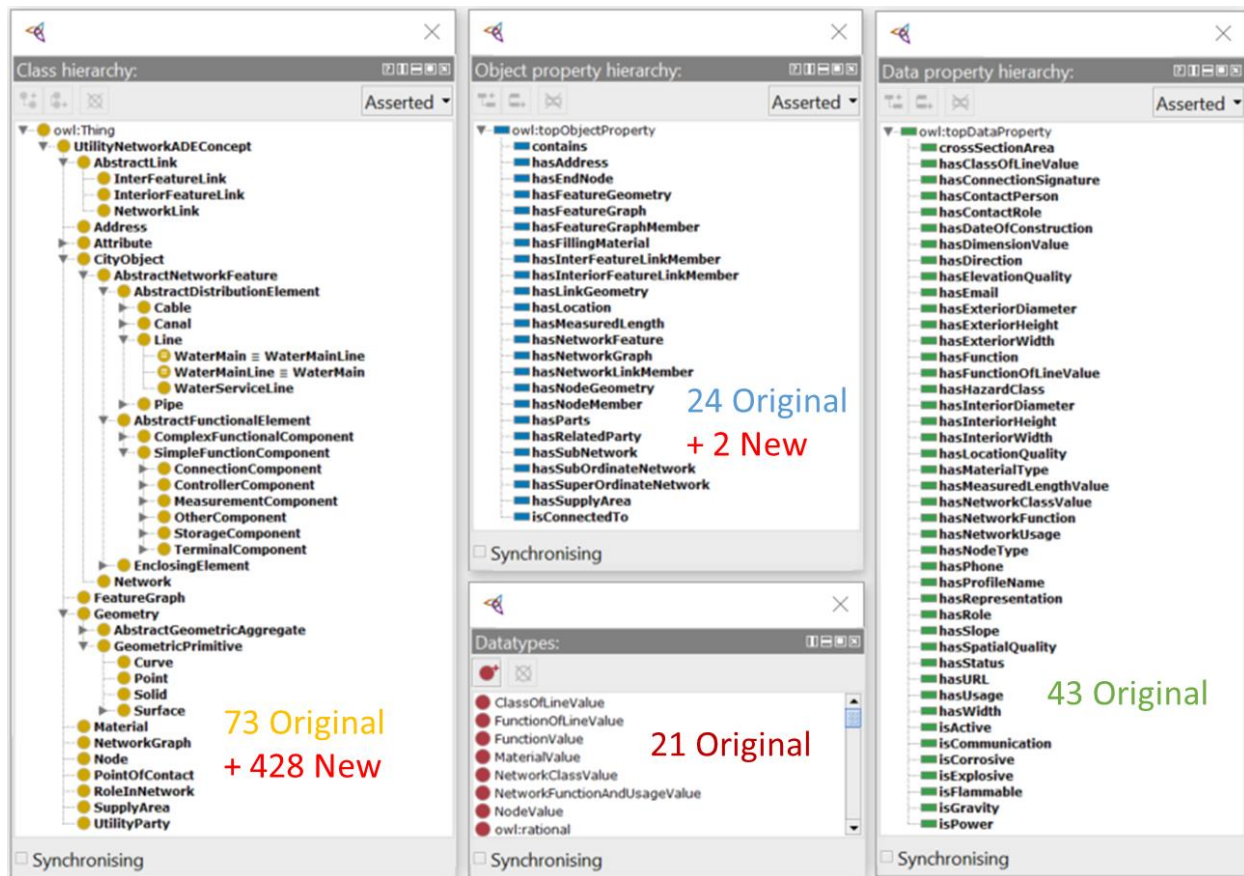


Figure 3.14. Hierarchies of the classes, object properties, data properties, and datatypes in the resulting ontology

The evaluation of the resulting ontology was conducted in an iterative manner, starting with evaluation through simple automated consistency checking to ensure correct syntax formalization (using OWL), followed by evaluation by domain experts. Preliminary consistency checks were successfully conducted through the built-in Protégé reasoner. As such, the ontology does not contain contradictory statements and are valid in the use of OWL syntax. Then, the ontology was assessed by domain experts to determine if they are accurate, sufficient and common conceptualization of the utility domain. Since (1) the base ontology inherits the semantics in the candidate open standard – CityGML Utility Network ADE, and (2) the ontology enrichment from glossaries achieves the maximum compliance with the base ontology, the majority of the detailed modeling in the resulting ontology was confirmed by the experts. After minor changes based on the feedbacks from experts, the revised ontologies were then fully agreed upon.

The following limitation is acknowledged. The ontology evaluation by domain experts did not involve enough industry experts, and the selection of the experts did not follow a strict screening process. In future, a sufficient number of industry experts will be selected based on the following criteria; 1) Years of experience within a particular sector of utility infrastructure, 2) Thorough knowledge of utility design issues, and 3) Familiarity with issues associated with design coordination among various utilities. Face-to-face interviews with experts will be conducted to assess the ontology contents (e.g., concepts and relationships) from the user's point of view. On average each interview will be designed to take one hour to complete. The experts will be briefed for 20 min about the sources of gathering different concepts and how they are structured to form hierarchies. Following the guidelines provided by EI-Diraby et al. [98], the experts will evaluate the ontology from the following aspects: navigational ease through locating concepts, categorizing concepts, and overall assessment.

- Navigational ease ensures knowledge access, retrieval, re-use, and maintenance. It is not difficult to locate concepts in an easy-to-navigate taxonomy hierarchy. Experts will be asked to locate ten concepts in the taxonomy with the definitions of those concepts given in the questionnaire to avoid any ambiguity. A six-point scale (1 being the easiest navigation and 6 being the most difficult navigation) will be used to record the experts' responses.
- Experts will be asked to rate their consensus with ten concepts regarding the classification in the ontology. Afterwards they will be presented with another set of ten concepts, till all the concepts in the ontology are rated. Likewise, experts will be asked to categorize these concepts according to the ontological model using the same six-point scale.
- Finally, as the experts become fully aware of the ontology and the conflicting needs of categorization, they will be asked to make a general assessment about the ontology still using a six-point scale.

Through survey analysis, the evaluation results by the experts can be used to modify the ontology and deliver a more acceptable outcome for the utility industry. Meanwhile, such evaluation will also provide a trustworthy assessment on the term extraction and semantic relationship classification results.

3.6 Summary and Conclusions

This paper develops a domain ontology for utility infrastructure by coupling the semantics of CityGML Utility Network ADE and domain glossaries. Departing from CityGML Utility Network ADE, a base ontology is developed through a series of UML-to-OWL mappings, followed by the incorporation of the semantics learned from domain glossaries. As domain glossaries are textual documents, an integrated NLP approach is devised to automatically learn the semantics from them. The NLP approach consists of a CRF model for term extraction and LSTM networks for semantic relationship classification. The learned semantics include a list of domain terms and semantically classified term pairs, which are then incorporated into the base ontology as new semantics. The proposed NLP approach was evaluated using human-annotated test sets, and results show an average accuracy of 96% in term extraction and 86% in semantic relationship classification. A case on the development of a domain ontology for utility infrastructure using CityGML Utility Network ADE and a glossary of water terms was demonstrated and the resulting ontology was evaluated to be an accurate, sufficient and shared conceptualization of the domain.

Unlike the traditional five-step approach that requires significant human efforts on knowledge retrieval, and ontology construction and validation, this paper takes a top-down strategy to develop the ontology for the utility infrastructure domain. An existing data model – Utility Network ADE was selected as the base, and thus significant time and effort were saved in building the knowledge skeletons of the ontology. Domain glossaries play critical roles in sharing and conveying domain knowledge and understanding. Full automation (using an integrated NLP approach) in ontology learning from domain glossaries is realized, thus enabling the automated enrichment of the ontology to keep up with the growth of new knowledge. Compared to the existing ontologies, the ontology developed in this paper is argued to be a better option as the interoperability facilitator for the utility domain attributed to the following two characteristics. First, the semantic schema in the new ontology aligns well with the CityGML extension. Plus, many GIS tools provide the capability of exporting data in the CityGML-compliant format. The new ontology can serve as an effective intermedium for the exchange of utility geospatial data in heterogeneous proprietary formats. Second, the semantic vocabulary in the new ontology has a relatively extensive coverage of concepts and relationships (which can also be expanded in timely and automated manners). It can help interpret the meaning of data and enable the semantic integration of data between disparate sources that use different sets of vocabularies. Therefore, the

new ontology can be utilized as the shared and reliable information source to facilitate a high degree of interoperability across the utility infrastructure domain.

4. ONTOLOGY AND RULE-BASED NATURAL LANGUAGE PROCESSING APPROACH FOR INTERPRETING TEXTUAL REGULATIONS ON UNDERGROUND UTILITY INFRASTRUCTURE

This chapter presents the design of an ontology- and rule-based NLP approach to automate the interpretation of utility regulations – extracting the requirements from the regulations and further formalizing them into logic clauses – for supporting automated compliance checking of underground utilities. The approach integrates ontologies to capture both domain and spatial semantics in utility regulations and encode pattern-matching rules for information extraction. An ontology- and deontic logic-based mechanism is also integrated to facilitate the semantic and logic-based formalization of utility-specific regulatory knowledge. The proposed approach was tested in interpreting the spatial configuration-related requirements in utility accommodation policies, and results show the newly developed approach achieves 94.7% recall and 98.2% precision in information extraction and 93.2% accuracy in information formalization.

In developing the proposed approach, the methods for semantic NLP-based information extraction by Zhang and EI-Gohary [49] and Zhou and EI-Gohary [50] were used and adapted to address the unique challenges in processing utility-specific regulations. The deontic logic (DL) representation by Salama and EI-Gohary [126] was also used and extended. Please see Section 4.2.2 for further details.

This work is under review in *Advanced Engineering Informatics*, 2020, Xin Xu and Hubo Cai. “*Ontology and Rule-based Natural Language Processing Approach for Interpreting Textual Regulations on Underground Utility Infrastructure*”. Table titles and figure captions have been modified to maintain the form of the dissertation.

4.1 Introduction

Underground utility infrastructure supports essential services such as water, gas, electricity, and telecommunication to the public. The physical complex networks share the underground space and must be spatially coordinated to ensure their performance and structural integrity [2,34]. Utility regulations stipulate the spatial configurations among underground utility networks and their surroundings to avoid interferences and disruptions of utility services [1,2,34]. In the current practice, practitioners perform compliance checking, with the aim of detecting violations in

designs and existing records, by manually going through the lengthy textual regulations, interpreting them subjectively based on their knowledge and experience, and checking massive and heterogeneous utility data against them [1,45]. This practice is neither efficient, nor sustainable, attributed to the large size of and the heterogeneity in utility regulatory documents [1] and the heavy reliance of the interpretation on human knowledge and subjective judgement – different interpreters might entail different meanings from the same clause [15]. Therefore, there is a critical need for an automated approach for the consistent interpretation of textual regulations on underground utilities to ensure the compliance of underground utility infrastructure.

A number of approaches have been attempted to automate the interpretation process for regulatory documents in the Architecture, Engineering, and Construction (AEC) domain. Examples include the use of hypertext and hypermedia to aid in navigating regulatory documents [46,47] and the use of document markup techniques to assist in analyzing the semantic structure of target regulatory requirements [48]. Nevertheless, these methods require intense manual efforts on annotating regulatory documents for further interpretation [24,48]. Natural Language Processing (NLP) methods have emerged in recent years to automate the extraction of requirements from textual documents such as building codes [49,50] and utility regulations [1]. Further, NLP has also been attempted to transform the extracted requirements into a structured format (i.e., logic clauses) for compliance checking [51]. Technical challenges in automating the interpretation of utility regulations include 1) heterogeneous technical terminologies – utility regulations contain a variety of technical terms since different disciplines and communities of practice may adopt different sets of vocabularies to describe their utility assets, and 2) the dominance of spatial constraints in utility regulations regarding location and clearance for the purposes of infrastructure safety, maintainability, and constructability, and public health and safety [2,34]. Consequently, a successful NLP method for the efficient and consistent interpretation of utility regulations must have the capacity to address the heterogeneity of technical terminologies and understand the spatial semantics from natural language.

Towards that end, this paper presents an ontology and rule-based NLP approach to automate the interpretation of utility regulations, i.e., extract the requirements from the regulations and formalize them into logic clauses, that can be further implemented in automated reasoning for utility compliance checking. The approach has the following specifics. Two ontologies have been developed: 1) urban product ontology (UPO) that covers the concepts related to urban physical

products and their varying names for capturing domain semantics from the heterogeneous terminologies in regulations and 2) spatial ontology (SO) that covers two layers of semantics – linguistic spatial expressions and formal spatial relations for understanding spatial language in regulations. A set of text patterns that consist of syntactic features (captured using common NLP techniques) and semantic features (captured using ontologies) have been defined and encoded as pattern-matching rules for information extraction. A mechanism by coupling ontologies and deontic logic (DL) has been designed to achieve the semantic and logic-based formalization of utility-specific regulatory knowledge, i.e., map the extracted information elements into their semantic correspondences and further transform them into DL clauses. The approach was tested in extracting and formalizing the spatial configuration-related requirements from utility accommodation policies. Results demonstrate its effectiveness as a means for the consistent and objective interpretation of textual regulations on underground utilities to ensure the compliance of underground utility infrastructure.

4.2 Background and Review of Related Studies

4.2.1 Automation in the interpretation of regulatory documents

With the advancements in computing technologies, a number of rule-based and automated methods for compliance checking have been developed in the AEC domain [15,19,20,24,33,117–121]. Despite this progressive trend, intensive manual efforts are still needed to interpret the regulatory documents and represent the requirements in a computable form [119]. A number of approaches have been taken by researchers to automate or semi-automate the interpretation process of regulatory documents. Examples include the use of hypertext and hypermedia to aid in navigating regulatory documents [46,47] and the use of document markup techniques to analyze the semantic structure of target regulatory requirements [48]. These efforts mainly focus on the analysis of the document structure, and thus, substantial manual efforts are still required to annotate regulatory documents for further interpretation [24,48].

In recent years, NLP methods have emerged as an effective tool to automate the interpretation process of textual documents. NLP deploys artificial intelligence to enable computers to understand, create, and analyze human languages [40]. It has been used in applications such as machine translation, speech recognition, information retrieval and information

extraction [114]. In the construction domain, a number of important research efforts have used NLP techniques for document classification [100,101,122,123], information retrieval [102–104,124], and information extraction [1,42,49,50,104,105,125]. For instance, Al Qady and Kandil [123] developed a text classifier to automatically classify project documents on the basis of text content; Zou et al. [103] utilized text mining and NLP techniques to retrieve similar cases from construction accident databases for risk management; and Zhang and EI-Gohary [49] proposed a semantic NLP approach to extract the requirements from building codes for supporting automated compliance checking.

NLP methods for the interpretation of regulatory documents for the purpose of compliance checking involve two steps: information extraction and information formalization. A number of studies have developed NLP methods to automate the extraction of requirements from textual documents such as building codes [49,50] and utility regulations [1]. NLP has also been attempted to transform the extracted requirements into logic clauses that could be directly used for automated compliance checking [51]. These studies have demonstrated the successful application of NLP-based approach in interpreting regulatory documents for compliance checking. However, existing NLP methods are highly domain specific and application dependent [126], and thus, the methods developed for the building sector are not suitable for the utility sector. While the method developed in [1] serves the utility domain, it was challenged by the lack of a comprehensive taxonomy to address the issue of heterogeneous terminologies and its limited capability of spatial understanding. Therefore, there is a need to improve existing NLP methods for the efficient and consistent interpretation of utility regulations to suit domain-specific application purposes (e.g., utility compliance checking).

4.2.2 NLP-based information extraction

NLP-based information extraction, one critical step towards the automated interpretation of textual regulations, aims to recognize meaningful information from unstructured data and formalize them in the structured/normalized format by deploying NLP techniques [114]. NLP-based information extraction mainly utilizes two approaches, a rule-based approach or a machine learning (ML) approach. The rule-based approach relies on pattern-matching rules for text processing. In most of rule-based information extraction systems [127], input texts are first processed as a sequence of tokens, human efforts are then involved in defining text patterns over

the text features of these tokens, and finally the defined patterns are encoded as pattern-matching rules for information extraction. The ML approach uses ML algorithms such as Naïve Bayes (NB), Support Vector Machines (SVM), Hidden Markov Models (HMM), or Conditional Random Fields (CRF) to automatically learn the extraction patterns/rules from a set of annotated training texts [128]. While this approach eliminates human involvement in text pattern definition and extraction rule development, it still requires human effort to prepare a sufficiently large size of training data.

In the construction domain, most of the NLP-based efforts adopted the rule-based approach to extract specific types of information based on partial analysis of textual documents. For instance, Abuzir and Abuzir [42] developed the ThesWB system which relied on the document structure and simple lexico-syntactic patterns to extract civil engineering terms and their relations from HTML web pages. Al Qady and Kandil [105] used limited syntactic features produced via shallow parsing to extract subjects, objects, and their relations from contract documents, and created a knowledge graph of the extracted information. Li et al. [1] utilized chunk-based rules to automatically extract information from utility regulations to support automated compliance checking of underground utilities. Lee et al. [129] integrated preprocessing, syntactic, and semantic rules to automatically extract poisonous clauses from international construction contracts. However, ML-based information extraction has been less studied in construction until recently; for instance, Liu and El-Gohary [130] developed a method of automated information extraction from bridge inspection reports based on CRFs. Kim and Chi [104] used an integrated approach – rule-based and CRF methods to automatically extract information from accident cases.

Recently, ontologies have been integrated to suit domain-specific information extraction purposes [52–55]. Ontology is an explicit and formal specification of a conceptualization [57], which allows for representing domain meanings in an information system. Ontology-based information extraction further incorporates semantic features into rule-based or ML-based systems to extract information based on meaning. It is reported that the use of ontology yields higher performance in information extraction for a specific domain [49,50,55]. For instance, Zhang and El-Gohary [49] proposed a semantic NLP-based approach, where extraction patterns are composed of a variety of syntactic and semantic features, to automatically extract information from building codes. Further, Zhou and El-Gohary [50] advanced the aforementioned approach through several domain-specific preprocessing techniques, a more complex extraction procedure, and a deeper domain ontology, to facilitate the information extraction from building energy conservation codes.

In developing the proposed approach, the methods for semantic NLP-based information extraction by Zhang and EI-Gohary [49] and Zhou and EI-Gohary [50] were used and adapted to address the unique challenges in processing utility-specific regulations. The methods were adapted in the following three ways:

- Using a different domain ontology for urban infrastructure – UPO. UPO captures the concepts related to utility and transportation physical products and also includes the heterogeneity of concept names. UPO-based information extraction allows for the extraction and formal representation of the varying technical jargons/terms from utility regulations.
- Using spatial language. Existing studies did not consider spatial cognition in their NLP algorithms. In this study, SO captures the concepts related to linguistic spatial expressions (that are used in spatial language) and formal spatial relations (that are used in spatial models). A set of spatial mappings are used to fill the semantic gap between linguistic spatial expressions and formal spatial relations. SO-based information extraction allows for the extraction and formal representation of spatial information from utility regulations.
- Using dedicated ontologies, UPO and SO, and DL for the semantic and logical formalization of utility-specific regulatory knowledge. The ontologies enable the semantic representation of the extracted information while DL provides a formal language with normative notions for the logical representation of the utility-specific regulatory knowledge. This study used the DL representation that was extended by Salama and EI-Gohary [126] for the accommodation of the regulatory requirements of automated compliance checking in the construction domain. Jointly, they facilitate the objective and consistent interpretation of textual regulations on underground infrastructure.

4.3 Ontology and Rule-based Approach for the Interpretation of Utility Regulations

An ontology and rule-based approach has been devised to automate the interpretation of utility regulatory documents. Figure 4.1 illustrates the connections among the composing elements and the workflow of the newly developed approach. It includes five major steps: 1) text preprocessing, 2) annotation of regulatory sentences, 3) analysis of target information elements, 4) extraction of target information elements, and 5) formalization of target information elements. Due to the dominance of the requirements regarding the spatial configurations between utilities and

their surroundings in utility regulations, this study focuses on the sentence-level interpretation of the spatial configuration-related requirements.

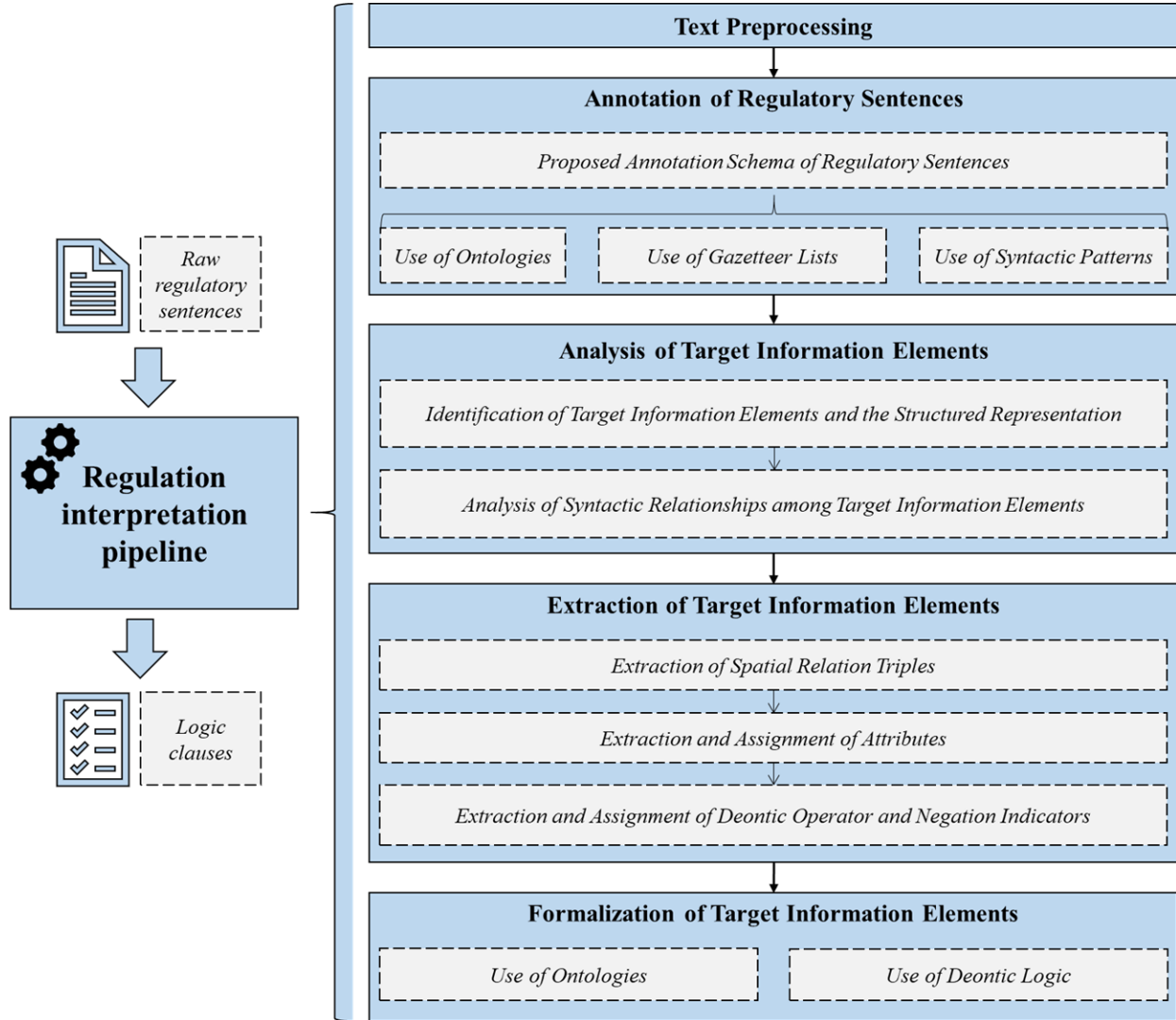


Figure 4.1. Proposed approach for the interpretation of utility regulations

4.3.1 Text preprocessing

Text preprocessing aims to extract the most basic syntactic features from input texts for subsequent NLP tasks. Preprocessing techniques used in this study include tokenization, sentence splitting, part-of-speech (POS) tagging, morphological analysis, and syntactic parsing. Many off-the-shelf tools now provide NLP pipelines for text preprocessing such as the ANNIE system of GATE [131]. Figure 4.2 presents an illustrative example of text preprocessing.

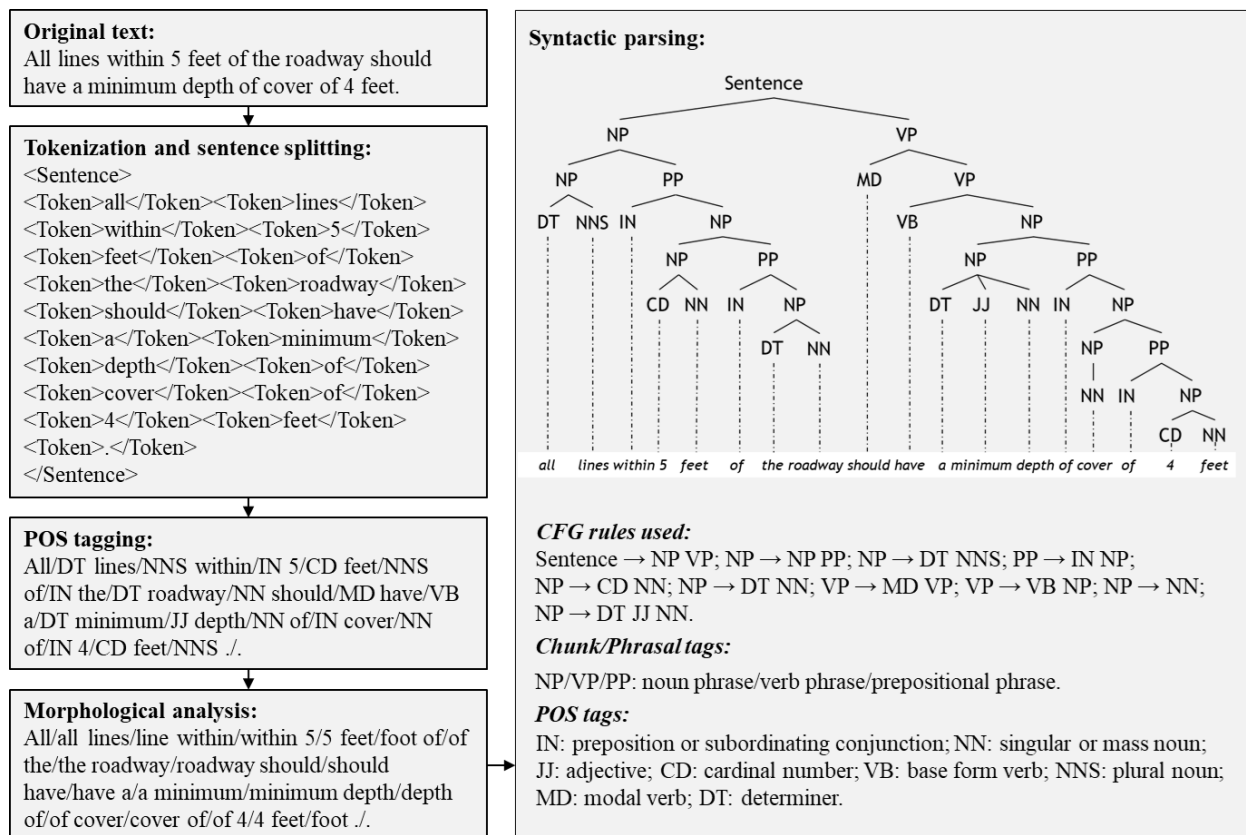


Figure 4.2. NLP pipeline for text preprocessing

In particular, syntactic parsing is the task of recognizing a sentence and assigning a syntactic structure to it according to context-free grammars (CFGs) [114]. CFGs define a set of rules to derive a tree structure from a given sentence. For instance, the CFG rule “Sentence → NP VP” indicates a sentence can be replaced as a combination of noun phrase and verb phrase. The derivation process continues until it reaches the individual word level. Syntactic parsing contributes to this study in two ways: 1) generates phrasal tags to capture more general text patterns and to reduce the possible number of enumerations in developing text patterns for extraction [49]; and 2) provides the syntactic structure of the complex sentence to support full sentence analysis for information extraction.

4.3.2 Annotation of regulatory sentences

In this step, an annotation schema was proposed to categorize different natural language expressions (such as a specific word, a phrase, or a chunk of text) in the regulatory sentences into

different annotation groups. Three main techniques (i.e., ontologies, gazetteer lists, and syntactic patterns) were used to enable the automated annotation of the regulatory sentences. This step aims to prepare intermediate text features (such as semantic features or application-specific features) from the preprocessed texts for the subsequent steps.

4.3.2.1 The annotation schema for regulatory sentences

Eight types of annotations are considered in the schema, i.e., spatial entity, spatial entity modifier, spatial lexical unit, deontic operator indicator, negation indicator, distance value, distance unit, and distance restriction. Figure 4.3 presents an example sentence annotated based on the schema.

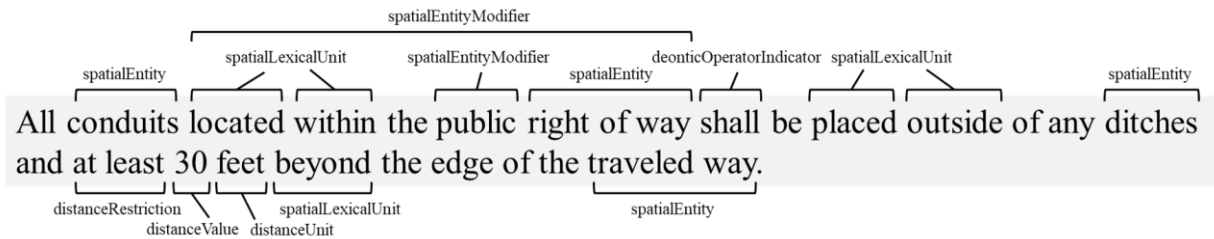


Figure 4.3. An annotated example of regulatory sentence

Specifically, “spatial entity” refers to the name/label of an urban product (e.g., conduit) whose location or position is described in the requirement; “spatial entity modifier” refers to a word or a chunk of words, such as adjectives (e.g., public), adverbs, or phrases (e.g., located within the public right of way), that give additional descriptions about the spatial entity; “spatial lexical unit” is the most basic lexical unit that has spatial implications (e.g., locate, within); “deontic operator indicator” is a word or phrase (e.g., shall) that indicates the deontic type of the requirement [49,50,126]: obligation, permission, or prohibition; “negation indicator” is the word “not” or “no”; “distance value” refers to the quantitative measure of the specified distance by the requirement (e.g., 30); “distance unit” refers to the measurement unit of the “distance value” (e.g., feet); “distance restriction” refers to the restriction set to the “distance value”, for specifying a quantitative range, such as “at least”.

4.3.2.2 Use of ontologies

Two ontologies, UPO and SO, were developed in this step to help annotate the spatial entities and spatial lexical units in the sentence. Figure 4.4 presents a partial view of the developed UPO and SO. UPO mainly captures the concepts related to urban physical products. It categorizes urban products into two main groups: utility products and transportation products. As for the fragmented utility industry, different sets of vocabularies are being used by different organizations for describing their owned products. For instance, different terms such as “cathodic protection anode bed”, “deep anode well”, and “deep ground bed” are often used to refer to the same utility product “cathodic protection well”. The existing ontologies for the utility infrastructure domain [2,34,132] primarily focus on concept description but neglect the heterogeneity of concept names and consequently, they are not sufficient to interpret the varying technical jargons/terms in utility regulations for the purpose of information extraction/formalization. In this study, UPO incorporates the term diversity and captures such diversity by assigning label property values to ontology concepts (see Figure 4.4). Thus, UPO-based annotation enables the annotation of spatial entities that have different names and meanwhile retain their correspondences to the ontology concepts.

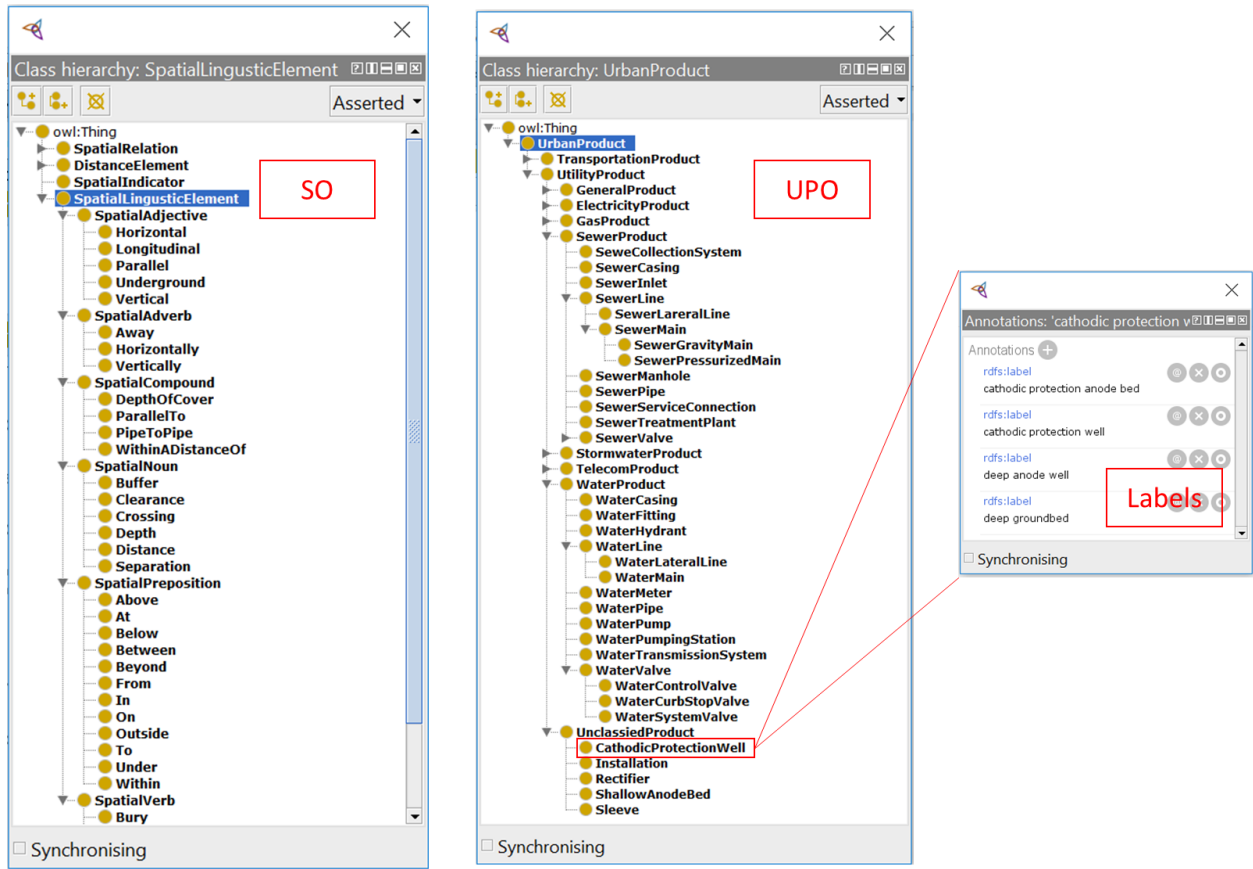


Figure 4.4. A partial view of SO and UPO

SO has four top-class concepts, i.e., spatial linguistic element, distance element, spatial indicator, and spatial relation. Existing spatial ontologies [45,63,133] merely consider the general concepts of spatial objects and spatial relations but do not include the linguistic spatial expressions that are used in natural language texts, thus preventing them from being applied in spatial information extraction from natural language texts. To the authors' best knowledge, SO is the first attempt that includes two layers of semantics – linguistic spatial expressions and formal spatial relations to allow for the extraction and formal representation of spatial information from utility regulations. This step mainly uses the concepts under the spatial linguistic element to help annotate the spatial lexical units in the sentence. Under this category, there are five subcategories: spatial nouns, spatial adjectives, spatial adverbs, spatial prepositions, and spatial compounds. Referring to SO, spatial lexical units within the sentence can be identified as well as their corresponding semantic categories.

The use of UPO and SO benefits the information extraction task in this study in two main ways: 1) creating a list of names and assigned labels of all ontology concepts in base forms as a semantic vocabulary for annotation lookup; and 2) enabling the awareness of the hierarchical relationship between super-sub concepts in defining text patterns for extraction [49,50]. The implementation is enabled through the OntoRoot Gazetteer module in GATE.

4.3.2.3 Use of gazetteer lists

A gazetteer is a set of lists storing specific terms that share a common category. Gazetteer lists have been used in previous information extraction efforts [1,49,50] to find occurrences of the stored terms in text. For this study, three gazetteer lists were manually compiled and used: 1) negation gazetteer list, which includes negation words like “no” or “not”; 2) distance unit gazetteer list, which includes unit words like “feet” and “inch”; 3) distance restriction gazetteer list, which is composed of words/phrases specifying a quantitative range of a quantity value, such as “at least”, “at most”, “minimum”, “greater or equal”, and “less than”. They were used to facilitate the automatic annotation of negation indicators, distance units, and distance restrictions in the sentences.

4.3.2.4 Use of syntactic patterns

Syntactic patterns were encoded as annotation rules in this step to enable sequential annotations among syntactically related annotations. For instance, as spatial entity modifiers are to give additional descriptions of the spatial entities, from the syntactic perspective, spatial entity modifiers could be the adjacent sentence constituents of the spatial entities, such as adjectives, adverbs, phrasal modifiers, or subordinate clauses. Regular expressions [114] were used to characterize possible combinations of text features (such as POS tags, phrasal tags, text strings, and existing annotations) for syntactic pattern matching. Java Annotation Patterns Engine (JAPE), a regular expression-based implementation in GATE, was used to encode pattern-matching rules for annotation. Figure 4.5 presents one example JAPE rule for annotating spatial entity modifiers. In this rule, JJ, JJR and JJS are POS tags for adjective, comparative adjective, and superlative adjective respectively; VP, PP, SBAR are phrasal tags for verb phrase, prepositional phrase, and subordinate clause respectively; and spatialEntity is the annotation for spatial entity. Using the

JAPE operators, patterns can be alternative (|), optional (?), or matched zero or more (*), one or more (+) or some specified number of times. By applying this rule to the example in Figure 3, adjectives (such as the word “public”) preceding the spatial entities and the followed verb phrase, preposition phrase or subordinate clause (such as the phrase “located within the public right of way”) would be annotated as spatialEntityModifier.

```
Rule: Annotate spatial entity modifiers based on the spatial entities
( ({{JJ}}|{{JJR}}|{{JJS}}):sem)* {spatialEntity} ({{VP}}|{{PP}}|{{SBAR}}):sem)? )
-->
:sem.spatialEntityModifier = {rule = "SentAnnot"}
```

Figure 4.5. An example JAPE rule

In addition, syntactic patterns reduce ambiguities during annotation. For instance, CD (a POS tag for cardinal number) tagged texts are potential distance values of regulatory requirements. However, they could also be quantity values that describe certain attributes of spatial entities such as dimensions. The syntactic closeness between the distance value and the spatial lexical unit is quite informative in reducing such ambiguity. For instance, in the sentence “the 6-inch mechanical joint inlet shall be located 5 feet below the ground”, “6” should not be annotated as distance value while “5” should be annotated due to its syntactic closeness to the spatial lexical units “located” and “below”.

Lastly, deontic operator indicators were annotated based on the MD (a POS tag for modal verb) tag. By following the above procedures, all annotations can be automatically added into the regulatory sentences.

4.3.3 Analysis of target information elements

This step aims to analyze the target information elements that need to be extracted from the sentences. First, the types of target information elements were identified based on the specific requirements of the application and the domain (i.e., spatial configuration-related requirements in utility regulations). A representation format was then proposed for structuring the identified information elements. Finally, the syntactic relationships among the target information elements were analyzed to provide guidelines on the extraction of these information elements.

4.3.3.1 Identification of target information elements and the structured representation

Seven types of target information elements that characterize the spatial configuration-related requirements were identified, i.e., “Trajector”, “Trajector attribute”, “Spatial indicator”, “Landmark”, “Landmark attribute”, “Deontic operator indicator”, and “Negation indicator”. Specifically, “Trajector” refers to the central object of the spatial configuration described in the requirement; “Spatial indicator” is the linguistic expression that signals the spatial relation in a spatial configuration; “Landmark” refers to the secondary object of the spatial configuration described in the requirement; “Trajector attribute” and “Landmark attribute” are the attributes of “Landmark” and “Trajector” respectively; “Deontic operator indicator” is the linguistic expression indicating the deontic type of the requirement: obligation, permission, or prohibition; “Negation indicator” is the negation word such as “not” and “no”.

A 7-tuple - <Trajector, Trajector attribute, Spatial indicator, Landmark, Landmark attribute, Deontic operator indicator, Negation indicator> - was proposed to structure the identified information elements. One 7-tuple represents one spatial configuration described in the requirements. For each regulatory sentence it may describe multiple spatial configurations using logic conjunctions (such as “and” and “or”) in different linguistic hierarchies (such as the main clause and the subordinate clause), and thus, the sentence-level regulatory information can be represented as hierarchically structured and logically connected (HSLC) 7-tuples. As shown in Figure 4.6, the regulatory information in the example sentence could be represented as one 7-tuple in the first hierarchy and two logically connected (using OR) 7-tuples in the second hierarchy. In this study, the first hierarchy refers to the main clause of a regulatory sentence while the second hierarchy refers to the modifying phrases or subordinate clause of the sentence.

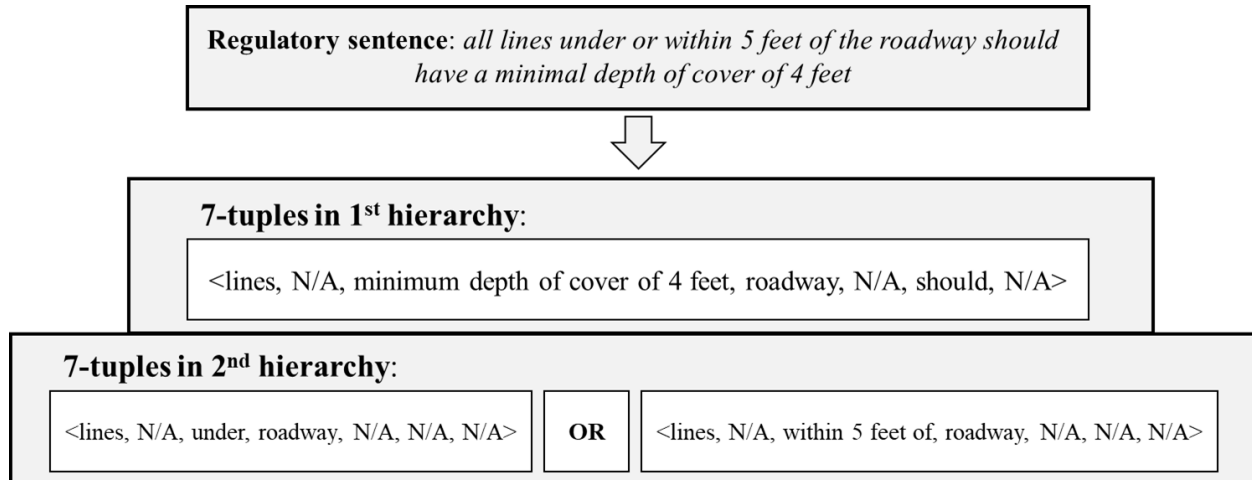


Figure 4.6. An example of regulatory sentence represented as HSLC 7-tuples

4.3.3.2 Analysis of syntactic relationships among target information elements

This section presents the analysis of the syntactic relationships among the 7-tuple elements. As the pivot of the 7-tuple, the spatial relation triple (SRT) - <Trajector, Spatial indicator, Landmark> - is the most basic unit to represent a spatial configuration; other tuple elements serve as additional descriptions to the SRT or its inside elements. As shown in Figure 4.7, the SRT relies on the expression grammar of the spatial indicator to link the trajector and the landmark; the trajector/landmark attribute are the syntactic modifiers of the trajector/landmark; the deontic operator and negation indicators share the same sentence hierarchy with the SRT. Figure 4.7 also presents the extraction bases (i.e., sentence annotations) of the target information elements. For instance, “spatial entity” is the extraction base of “Trajector” and “Landmark”, which means, the texts annotated as “spatial entity” could be the potential instances of “Trajector” and “Landmark”. Together with the syntactic analysis results, text patterns can be defined to extract the target information elements and subsequently structure them as HSLC 7-tuples.

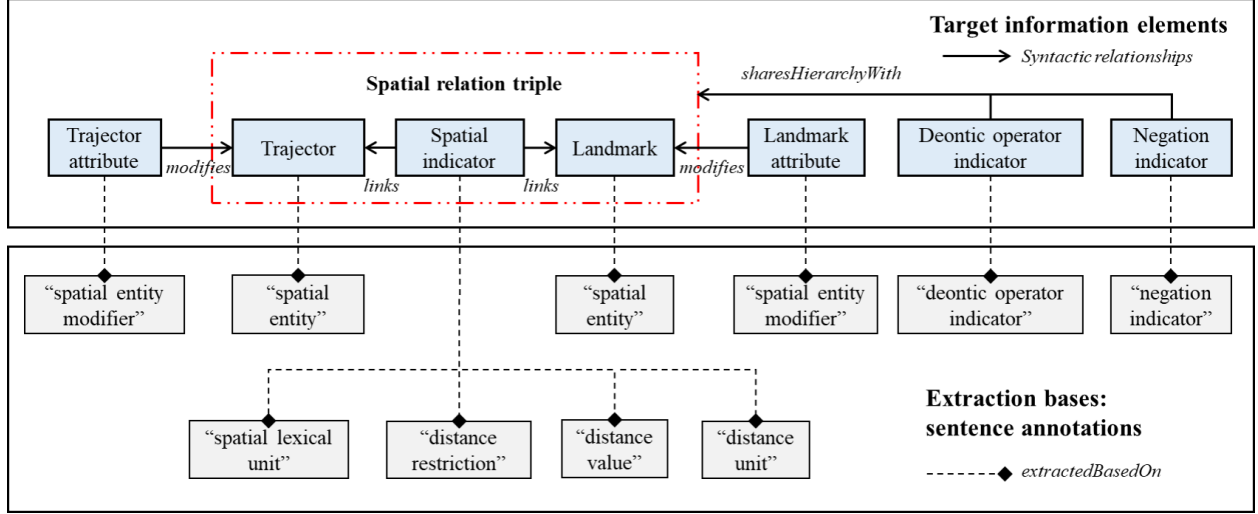


Figure 4.7. The syntactic relationships among the 7-tuple elements and the extraction bases

4.3.4 Extraction of target information elements

This step aims to extract the target information elements and structure them as HSLC 7-tuples. The extraction process follows the order of 1) extraction of SRTs, 2) extraction and assignment of attributes, and 3) extraction and assignment of deontic operator and negation indicators.

4.3.4.1 Extraction of SRTs

This section presents the process of extracting trajectors, spatial indicators, and landmarks from the sentences and organizing them into HSLC SRTs.

(1) Define text patterns for spatial indicators and SRTs

From Figure 4.7, sentence annotations of “spatial lexical unit”, “distance restriction”, “distance value”, and “distance unit” serve as the main bases to extract the spatial indicators. Examining through the development set (i.e., a collection of regulatory sentences that are used for determining text patterns), the combination patterns of those extraction bases were defined for spatial indicators. Table 4.1 lists several typical text patterns for spatial indicators and their corresponding matched texts in the sentences. For instance, a sequence of past participles of spatial verbs (optional), distance elements (optional), spatial adverbs (optional), and spatial prepositions, denoted as $(\{SV, VBN\})?(\{D\})?(\{SAdv\})?\{SP\}$, matches the spatial indicators in the sentences

such as “above”, “situated below”, “placed 5 feet under”, “located at least 10 feet from”, “one foot vertically above”, etc. Table 4.1 also lists several expression patterns for the SRTs corresponding to the spatial indicators. For instance, a sequence of “Trajector”, “Spatial indicator”, and “Landmark”, denoted as {T} {SI} {L}, is a type of SRT expression pattern. Matched texts include “conduits located within the public right of way”, “water mains crossing other utilities”, and “water lines installed within 5 feet of the roadway”.

Table 4.1. Example patterns for SI and SRT and their corresponding matched texts

Text patterns for SI	$((\{SV, VBN\})? (\{D\})? (\{SAdv\})? \{SP\})$	$((\{SV, VBN\})? \{SP, \text{“within”}\} \{D\} \{\text{“of”}\})$	$((\{SV, VBN\})? \{SC, \text{“within a distance of”}\} \{D\} \{SP\})$	$\{SV\} (\{SP\})?$
Matched texts of SI	above, situated below, placed 5 feet under, located at least 10 feet from, one foot vertically above, etc.	within 5 feet of, installed within 25 horizontal feet of, etc.	located within a distance of 300 meters below, etc.	cross, crossing, cross above, run through, intersecting, touch, containing, etc.
Expression patterns for SRT	$\{T\} \{SI\} \{L\}$			
Matched texts of SRT	conduits located within the public right of way, supply lines placed one foot vertically below any water main, etc.	water lines installed within 5 feet of the roadway, etc.	utility assets within a distance of 30 feet beyond the travelled way, etc.	water mains cross over sewer mains, water mains crossing other utilities, etc.
<i>--Continued--</i>				
Text patterns for SI	$((\{DR\})? (\{SA\})? \{SN\} (\{\text{“of”}\} \{D\})? \{SV\} (\{SAdj\})?)$			
Matched texts of SI	minimum vertical clearance of 18 inches, minimum depth of 4 feet, horizontal separation, minimum cover, etc. cross, intersect, run parallel, etc.			
Expression patterns for SRT	$\{SI\} \{\text{“between”}\} \{T\} \{\text{“and”}\} \{L\}, \{SI\} \{\text{“for”}\} \{T\} \{\text{implicit L}\} \{T\} \{\text{“and”}\} \{L\} \{SI\}$			
Matched texts of SRT	a horizontal separation of 10 feet between water mains and sewer mains, minimum depth for sewer mains, etc. gas lines and sewer lines intersect, gas lines and sewer lines run parallel, etc.			

Note: SO concepts: SV – spatial verb, SP – spatial preposition, SAdv – spatial adverb, SAdj – spatial adjective, SC – spatial compound, SN – spatial noun; Sentence annotations: DR – distance restriction, DV – distance value, DU – distance unit; Target information elements: T – trajector, SI – spatial indicator, L – landmark; D – distance, i.e., ({DR})? {DV}{DU}.

(2) Define pattern-matching rules for SRT extraction

Since the vast majority (over 75%) of spatial indicators have the grammar pattern of {T}{SI}{L} to express SRTs in the sentences, this section presents the process of extracting SRTs from the sentences based on this representative expression pattern, as shown in Figure 4.8.

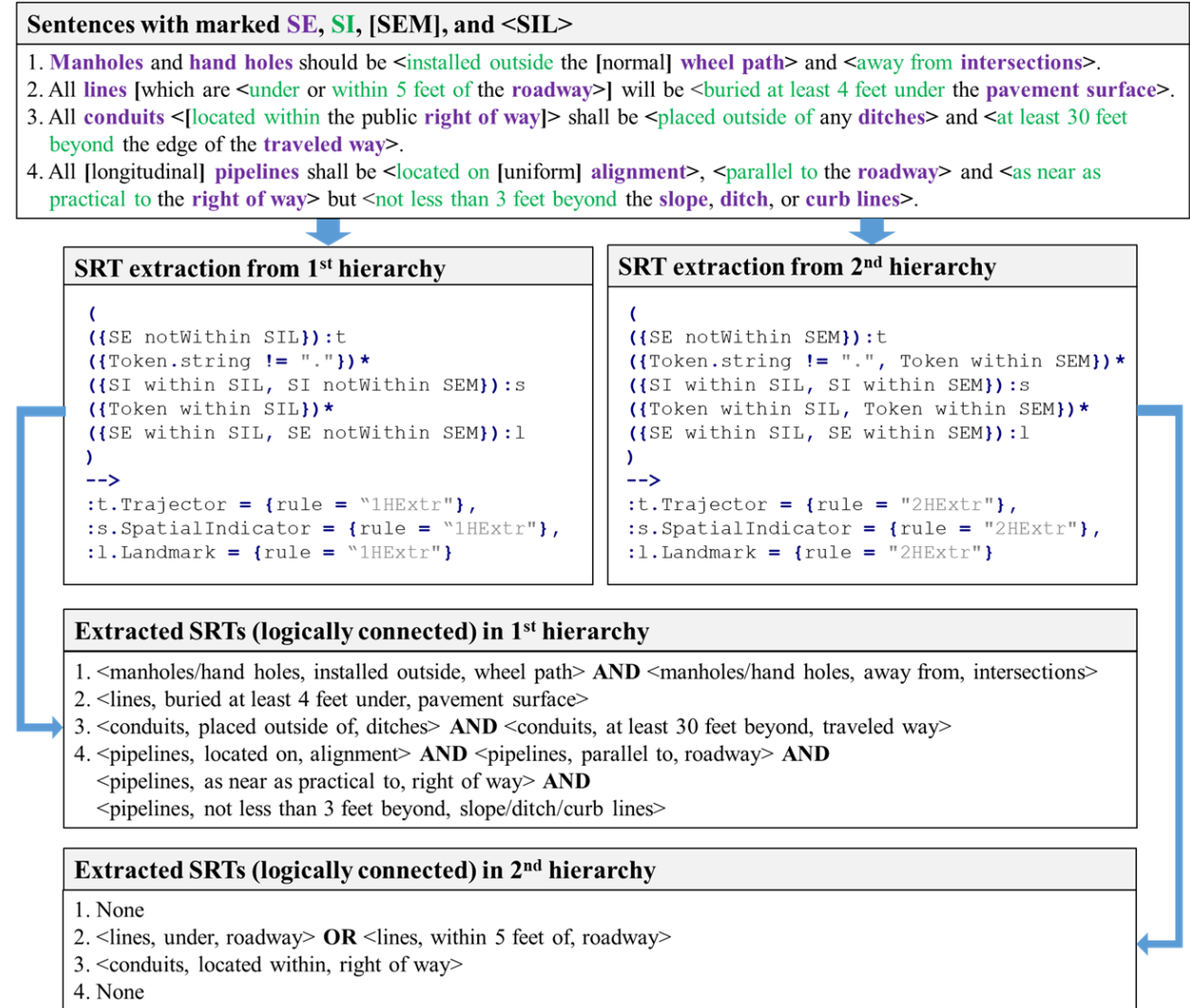


Figure 4.8. Extraction process of HSLC SRTs

Given the fact that there could be multiple spatial entities and spatial indicators in one single sentence (such as the example sentences in Figure 4.8), which may result in multiple

concurrent SRTs in the same sentence, all possible combination patterns were modeled in the pattern-matching rules to extract these triples simultaneously. It was also observed at the first attempt that these rules would fail to extract SRTs from the sentences that have different hierarchies (such as sentences No. 2 and No. 3 in Figure 4.8). As such, extraction was performed separately from different hierarchies. As shown in Figure 4.8, different hierarchies have their corresponding SRT extraction rules. Specifically, the following JAPE rule was encoded to extract SRTs from the first hierarchy (i.e., the main clause of the sentence): “({SE notWithin SIL}):t ({Token.string != “.”})* {SI within SIL, SI notWithin SEM}:s ({Token within SIL})* ({SE within SIL, SE notWithin SEM}):l --> t.Trajector, s.SpatialIndicator, l.Landmark”. In this rule, SE, SI, and SEM refer to the spatial entity, spatial indicator, and spatial entity modifier, respectively; SIL is an intermediate pattern that matches the combination of paralleling spatial indicators and spatial entities (potential landmarks), such as “under the roadway”, “under or within 5 feet of the roadway”, “3 feet beyond the slope, ditch, or curb lines”; within/notWithin is the contextual operator implemented in JAPE to match annotations within the context of other annotations. By specifying the control style as all, this JAPE rule will trigger all matching patterns to extract SRTs with all kinds of combination patterns. The extraction results are also given in Figure 4.8. Following the same procedure, the SRTs in the second hierarchy can be extracted. Once all SRTs are extracted from different hierarchies, the last step is to determine the logic connections among them. The conjunctions used in the sentences serve as the main basis for the determination. For instance, in sentence No.2, two concurrent SRTs were extracted from the second hierarchy and then connected using the operator OR since the conjunction “or” is used in the sentence.

While Figure 4.8 illustrates the SRT extraction process using the {T}{SI}{L} pattern as the example, the same procedure applied to other patterns such as {T}{“and”}{L}{SI} and {SI}{“between”}{T}{“and”}{L} for extracting their corresponding SRTs from the sentences.

4.3.4.2 Extraction and assignment of attributes

This step aims to extract the attributes for the trajectors and landmarks and assign them to their corresponding SRTs. From Figure 4.7, spatial entity modifiers are the extraction bases of trajector and landmark attributes. Among the spatial entity modifiers there could be some related to spatial configurations, which have been handled during the previous step. This step mainly focuses on the extraction of non-spatial attributes such as dimensions and material types. Based

on this, the texts annotated as “spatial entity modifier” but without spatial implications would be the potential instances of trajector and landmark attributes. For example, in the sentence “the 6-inch mechanical joint inlet shall be located 5 feet below the ground”, “6-inch”, as the modifier to “mechanical joint inlet”, was extracted as the trajector attribute while no modifier was found for “ground”, thus, there was no landmark attribute. The resulting tuple with the assigned attribute would be <mechanical joint inlet, 6-inch, located 5 feet below, ground, N/A>.

4.3.4.3 Extraction and assignment of deontic operator and negation indicators

This step aims to extract the deontic operator and negation indicators and assign them to their corresponding SRTs. From Figure 4.7, the texts annotated as “deontic operator indicator” and “negation indicator” can be directly extracted as the instances of “Deontic operator indicator” and “Negation indicator”, respectively. The extracted information elements are then assigned to the SRTs based on the shared sentence hierarchies. For the sentence No.3 in Figure 4.8, the deontic operator indicator “shall” belongs to the first hierarchy and accordingly, “shall” is assigned to the SRTs in the same hierarchy. Since no negation indicator was found, there would be no negation indicator assigned to the SRTs. Following this, together with the assignment of attributes, the extracted SRTs can be expanded as HSLC 7-tuples. If no attributes, deontic operator indicators, or negation indicators were found or no assignments were made, their corresponding information elements in the 7-tuples would remain empty. Therefore, for the sentence No.3, the resulting HSLC 7-tuples would be:

- First hierarchy: <conduits, N/A, placed outside of, ditches, N/A, shall, N/A> AND <conduits, N/A, at least 30 feet beyond, traveled way, N/A, shall, N/A>.
- Second hierarchy: <conduits, N/A, located within, right of way, public, N/A, N/A>.

4.3.5 Formalization of target information elements

This step aims to formalize the extracted HSLC 7-tuples into logic clauses. Two specific tasks were involved in this step: 1) semantic formalization of the 7-tuple elements including the trajectors/landmarks, their attributes, and spatial indicators, and 2) logic representation of the HSLC 7-tuples.

4.3.5.1 Semantic formalization via ontologies

In this task the trajectors/landmarks, their attributes are mapped to their semantic correspondences in UPO. Since UPO concepts were retained as the semantic features during the annotation of spatial entities, and the extracted trajectors/landmarks inherited these semantics from the spatial entities, the trajectors/landmarks were mapped to their corresponding UPO concepts according to the semantic features. Certain attributes are also modeled in UPO. For example, `hasMaterialType` is a UPO relationship used for describing the material types of urban products. Based on the modeled relationships, trajector/landmark attributes can be mapped to their corresponding representations in UPO. For example, the trajector attribute “6-inch” for the trajector “mechanical joint inlet” was formalized as `MechanicalJointInlet(X)` and `hasDimension(X, 6-inch)`, where `MechanicalJointInlet` is a UPO concept and `hasDimension` is a UPO relationship.

Spatial indicators are formalized as spatial relations via SO. Figure 4.9 illustrates the relationships among the top-level concepts in SO: spatial linguistic elements and distance elements work together to form the spatial indicators that indicate spatial relations. The *indicate* relationship in SO is critical to the formalization of spatial indicators. However, spatial indicators are usually natural language expressions, which may be subjective for human interpretation, and thus, there is a semantic gap between the spatial indicators and their formal indications.

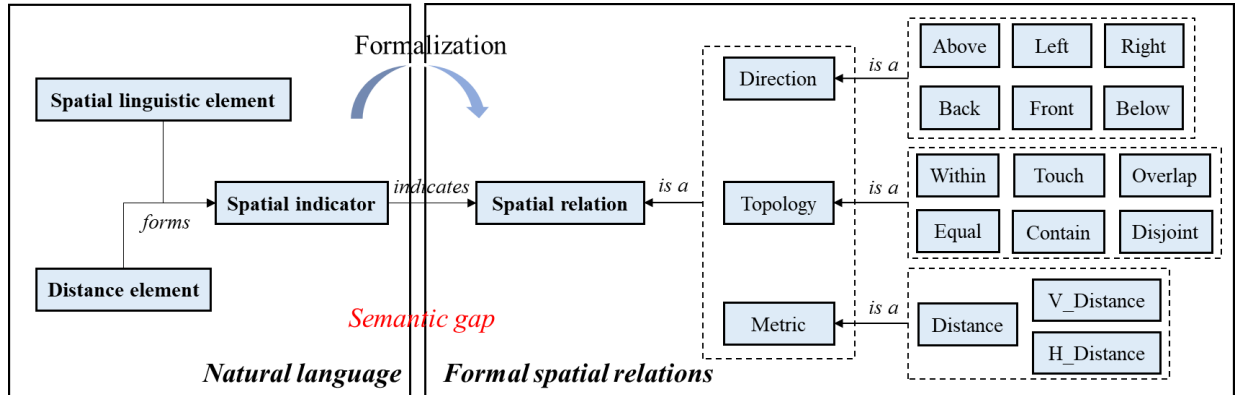


Figure 4.9. The relationships among the SO concepts and the formal spatial relations

Two types of mappings – pre-learned and hand-crafted – were used to map spatial indicators to spatial relation types. The pre-learned mapping relies on the connections between the spatial language (mostly spatial prepositions) to formal spatial relations learned through ML [133], e.g., the match between the spatial prepositions of “under” or “below” to the spatial relation of

“Below”. The hand-crafted mapping is domain-specific. For example, the spatial compound “depth of cover” is mapped to two spatial relations “Below” and “Distance” and the combination of spatial adjective and spatial noun “horizontal separation/clearance” is mapped to the spatial relation “H_Distance”.

Besides, spatial indicators are composed of spatial linguistic elements (e.g., spatial prepositions, spatial nouns) and distance elements (e.g., distance restrictions, distance values). These composing elements may have their respective spatial indication, and one spatial indicator can be mapped to multiple spatial relations. To address this issue, for every spatial indicator the composing elements were mapped to their corresponding spatial relations to collectively represent the original spatial indicator. For example, the spatial indicator of “at least 18 inches vertically above” results in two spatial relations. The spatial preposition “above” corresponds to the spatial relation “Above”, represented as $\text{Above}(\text{TrajectorX}, \text{LandmarkX})$. The spatial adverb “vertically” corresponds to the spatial relation “V_Distance”, which is further specified by “at least”, “18”, and “inches” and represented as $\text{V_Distance}(\text{TrajectorX}, \text{LandmarkX}, \text{inch}) \geq 18$.

4.3.5.2 Logic representation via deontic logic

DL is an extension of first order logic (FOL) to include normative notions for the formal representation and specification of laws, legal rules, and precedents [134]. DL is viewed as the most promising formal specification language for legal contracts [135]. DL is useful for representing utility regulations because its basic notations are fundamental for capturing the normative modalities of the requirements (e.g., what is obligated, what is permitted, and what is forbidden).

A DL statement consists of a set of predicates or functions that are combined or specified using two types of operators: deontic modal operators (i.e., obligation “ O ”, permission “ P ”, and prohibition “ F ”) and FOL operators (i.e., conjunction “ \wedge ”, disjunction “ \vee ”, negation “ \neg ”, and implication “ \supset ”). In addition, quantifiers (i.e., “ \forall ” and “ \exists ”) are also used to make assertions about the variables in DL statements. The logic representation of the HSLC 7-tuples in DL is described as follows.

- Formalized trajectors/landmarks and their attributes correspond to the predicates in DL statements. For example, $\text{WaterLine}(X)$ and $\text{hasDimension}(X, 6\text{-inch})$ are two predicates:

WaterLine and hasDimension are predicate symbols (representing UPO concepts or relationships) while “X” and “6-inch” are arguments, where “X” is a variable.

- For the formalized spatial indicators, topology and direction relations correspond to the predicates (e.g., Above(X, Y), where “X” and “Y” are arguments/variables) while metric relations correspond to the functions (e.g., Distance(X, Y, inch) = d, where “X” and “Y” are arguments/variables, “inch” is an argument, and “d” is the returned value) in DL statements.
- Predicates/functions corresponding to the spatial indicators that pertain to the first-hierarchy tuples appear in the RHS (succeeding the operator “ \supset ”) while the remaining ones appear in the LHS. On either side, predicates/functions generated from the same tuple are combined using conjunction “ \wedge ” while the combination of those from different tuples is determined based on the specific connections (i.e., “OR” or “AND”) among these tuples.
- Deontic operator indicators correspond to the deontic modal operators, which are used to specify the normative modalities of certain predicates/functions in DL statements. For example, “shall” corresponds to obligation “O”, thus, O(Above(X, Y)) means that Above(X, Y) is obligated as per requirements.

The following presents the resulting DL statements for the sentences No.2 and No.3 in Figure 4.8.

- Sentence No.2: $\forall x, y, z (Pipeline(x) \wedge Roadway(y) \wedge PavementSurface(z) \wedge (Below(x, y) \vee Distance(x, y, foot) < 5)) \supset O(Below(x, z) \wedge Distance(x, z, foot) \geq 4)$
- Sentence No.3: $\forall x, y, z, h (Pipeline(x) \wedge RightOfWay(y) \wedge Within(x, y) \wedge Ditch(z) \wedge Roadway(h)) \supset O(Disjoint(x, z) \wedge Distance(x, h, foot) \geq 30)$

While DL formalization supports automated reasoning in compliance checking, presently no deontic reasoner has been developed yet [126,136,137]. An alternative is to translate DL clauses into SPARQL queries for utility compliance checking in the context of semantic web. SPARQL suits the needs that are unique to utility compliance checking due to its capabilities of semantic understanding and spatial extension [95]. For the illustration purpose, the corresponding SPARQL queries for sentence No.3 for detecting utility noncompliance is spelled out below.

SELECT ?x WHERE { ?x a upo:Pipeline. ?y a upo:Ditch. ?z a upo:Roadway. ?h a upo:RightOfWay. ?x function:Within ?h. NOT EXISTS { ?x function:Disjoint ?y} UNION FILTER (function:Distance(?x, ?z, foot) < 30). }

4.4 Implementation

The proposed approach includes information extraction and formalization. Information extraction was implemented in GATE by configuring the following built-in/plugin tools: ANNIE English Tokenizer, ANNIE Sentence Splitter, ANNIE POS Tagger, GATE Morphological Analyzer, Stanford Parser, ANNIE Gazetteer, OntoRoot Gazetteer, and JAPE Transducer. For this study, ontologies were input into the OntoRoot Gazetteer module; gazetteer lists were added to the ANNIE Gazetteer module; and pattern-matching rules were input into the JAPE Transducer for sentence annotation and information extraction. Figure 4.10 presents the implementation architecture. GATE outputs an XML document that contains all added annotations along with their corresponding features.

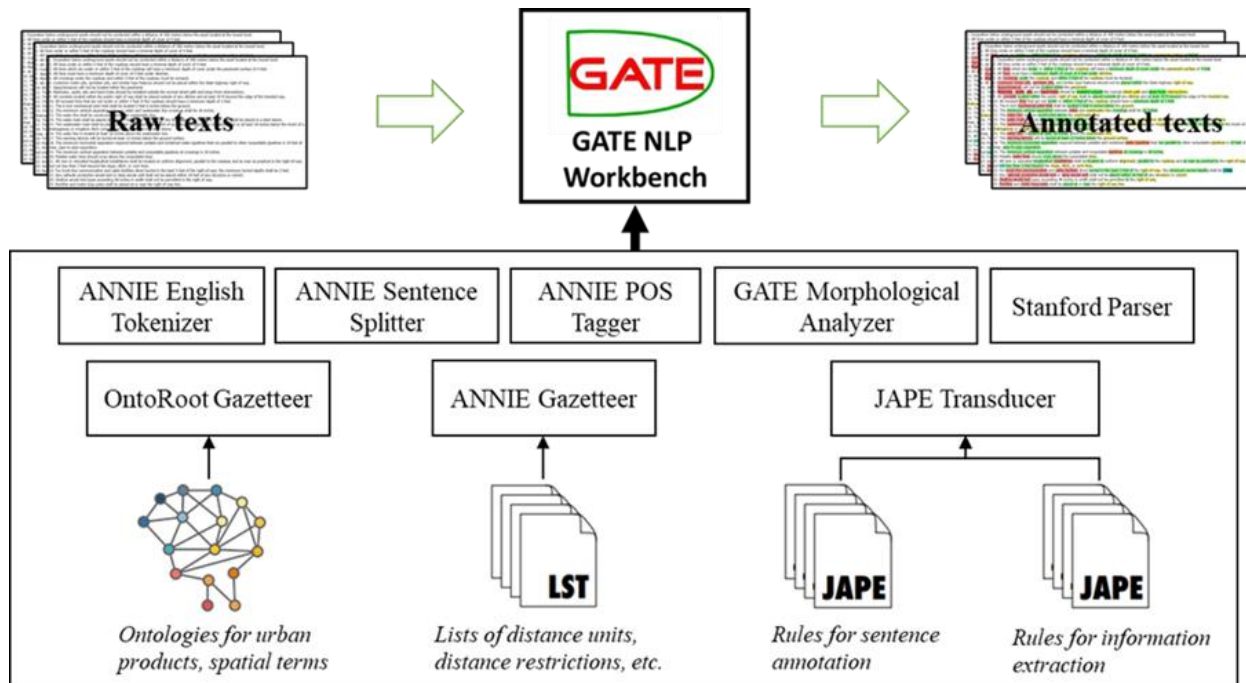


Figure 4.10. Implementation architecture for information extraction

A *Python* program that consists of three subprograms - XML parser, semantic mapper, and DL constructor - was developed to process the GATE XML document for information

formalization. The XML parser outputs the HSLC 7-tuples based on the XML annotations and features; the semantic mapper maps the 7-tuple elements to their semantic formalizations using UPO and pre-learned/hand-crafted spatial mappings; and the DL constructor finalizes the logical representation of the HSLC 7-tuples.

Figure 4.11 presents three representative examples of the requirements (each corresponds to one deontic type – obligation, prohibition, and permission) to illustrate their corresponding processes of information extraction and formalization as well as the future application – using SPARQL queries for utility compliance checking.

1. Original Text			
Normative modality	Obligation	Prohibition	Permission
Regulatory requirement	All conduits located within the public right of way shall be placed outside of any ditches and at least 30 feet beyond the edge of the traveled way.	Any deep anode well shall not be placed within 20 feet of any culvert.	Irrigation pipes crossing state right of way may be permitted.

2. Text Preprocessing			
Syntactic features: POS tags, phrasal tags, and syntactic structures. Example output: see Figure 2			

3. Annotation of Regulatory Sentences			
Annotation types: spatial entity, spatial entity modifier, spatial lexical unit, deontic operator indicator, negation indicator, distance value, distance unit, and distance restriction. Annotation techniques: UPO for spatial entity; SO for spatial lexical unit; gazetteer lists for negation indicator, distance unit, and distance restriction; syntactic patterns (encoded in JAPE) for spatial entity modifier, deontic operator indicator, and distance value. Example output: see Figure 3			

4. Analysis of Target Information Elements			
Target requirements: spatial configuration-related requirements in utility regulations. Target information elements: “Trajector”, “Trajector attribute”, “Spatial indicator”, “Landmark”, “Landmark attribute”, “Deontic operator indicator”, and “Negation indicator”. Structured representation: HSLC 7-tuples (see Figure 6). Extraction guidelines: syntactic dependencies among the target information elements and their extraction bases (see Figure 7)			

5. Extraction of Target Information Elements			
Text patterns for spatial indicators and spatial relation triples:			
Text patterns for SI	{(SV, VBN)}? {(D)}? {(SAdv)}? {SP}	{(SV, VBN)}? {SP, “within”} {D} {“of”}	{SV} {(SP)}?
Expression patterns for SRT	{T} {SI} {L}		
Pattern-matching rules (encoded in JAPE) for extracting HSLC 7-tuples: HSLC SRTs are first extracted from different sentence hierarchies (see Figure 8 for example JAPE rules), followed by the extraction and assignment of attributes, deontic operator and negation indicators.			
Extraction results:			
7-tuples in 1 st hierarchy	<conduits, N/A, placed outside of, ditches, N/A, shall, N/A> AND <conduits, N/A, at least 30 feet beyond, traveled way, N/A, shall, N/A>.	<deep anode well, N/A, placed within 20 feet of, culvert, N/A, shall, not>	<irrigation pipes, N/A, crossing, right of way, state, permitted, N/A>
7-tuples in 2 nd hierarchy	<conduits, N/A, located within, right of way, public, N/A, N/A>.	N/A	N/A

6. Formalization of Target Information Elements			
Formalization of trajectors/landmarks, their attributes, and spatial indicators (using ontologies):			
Formalization of trajectors/landmarks and their attributes	conduits → Pipeline(x); ditches → Ditch(y); traveled way → Roadway(z); right of way → RightOfWay(h)	deep anode well → DeepAnodeWell(x); culvert → Culvert(y)	irrigation pipes → IrrigationPipe(x); right of way, state → StateRightOfWay(y)
Formalization of spatial indicators	placed outside of → Disjoint(x, y); at least 30 feet beyond → Distance(x, z, foot) ≥ 30; located within → Within(x, h)	placed within 20 feet of → Distance(x, y, foot) < 20	crossing → Overlap(x, y)
Formalization of HSLC 7-tuples (using DL):			
DL clauses	$\forall x, y, z, h (Pipeline(x) \wedge RightOfWay(h) \wedge Within(x, h) \wedge Ditch(y) \wedge Roadway(z)) \supset O(Disjoint(x, y) \wedge Distance(x, z, foot) \geq 30)$	$\forall x, y (DeepAnodeWell(x) \wedge Culvert(y)) \supset F(Distance(x, y, foot) < 20)$	$\forall x, y (IrrigationPipe(x) \wedge StateRightOfWay(y)) \supset P(Overlap(x, y))$

7. Future Application – Utility Compliance Checking			
SPARQL queries for automated reasoning:			
SPARQL queries	SELECT ?x WHERE { ?x a upo:Pipeline. ?y a upo:Ditch. ?z a upo:Roadway. ?h a upo:RightOfWay. ?x function:Within ?h. NOT EXISTS { ?x function:Disjoint ?y } UNION FILTER (function:Distance(?x, ?z, foot) < 30). }	SELECT ?x WHERE { ?x a upo:DeepAnodeWell. ?y a upo:Culvert. FILTER (function:Distance(?x, ?y, foot) < 20). }	SELECT ?x WHERE { ?x a upo:IrrigationPipe. ?y a upo:StateRightOfWay. ?x function:Overlap ?y. }

Figure 4.11. Illustrative examples of information extraction and formalization

4.5 Experiments and Results

4.5.1 Experiment setup – source text selection and ontology development

Utility accommodation policies, such as INDOT Utility Accommodation Policy [138] and GDOT Utility Accommodation Policy and Standards [139], were selected because they contain numerous textual descriptions of the spatial configurations among utilities and their surroundings. A total of 300 sentences that contain spatial configuration-related requirements were collected. For each sentence, the target information elements and information formalization results (i.e., semantic correspondences and logic clauses) were also manually documented to form the ground truth for this experiment.

The ontologies (i.e., UPO and SO) used in this experiment were developed following the four-step procedure: 1) purpose and scope definition, 2) taxonomy building, 3) relation modeling, and 4) ontology coding [86]. UPO aims to capture the main concepts related to urban products while SO aims to capture main concepts related to linguistic spatial expressions and spatial relations. The resulting UPO covers a total of 312 concepts (along with 472 term labels assigned to the concepts) while the SO covers a total of 69 concepts (including 44 concepts of spatial linguistic elements). In UPO, relationships (such as *hasDimension* and *hasMaterialType*) are modeled to describe the attributes of urban products, which facilitate the semantic formalization of the extracted attributes from text. In SO, relationships are modeled to describe the semantic links between the spatial concepts, which facilitate the spatial understanding from natural language. UPO and SO are coded in the Web Ontology Language (OWL) format.

4.5.2 Development of text patterns for information extraction

200 sentences were randomly selected as the development set while the rest serve as the test set. The development set was manually annotated and then processed to generate the syntactic and semantic text features. The scrutiny of the hand-annotated ground truth led the authors to define the patterns for annotation over the text features. Table 4.2 presents the number of defined annotation patterns for the development set. The use of ontologies, gazetteer lists, and phrasal tags reduces the possible number enumerations in defining patterns. For instance, the numbers of patterns for annotating spatial entities and spatial lexical units were downsized to 1 and 6, respectively, by merely using the super-concepts in UPO/SO to cover all sub-concepts. Once the

sentences were annotated, text patterns for extracting the target information elements were defined. The process was conducted in an iterative manner. A preliminary set of extraction patterns were first hand-crafted and then applied back to the development set. If the extraction results are not satisfactory, the process may be iterated for performance improvement, resulting in additional extraction patterns in the pattern target set. The pattern set may be considered as final if the performance is satisfactory compared to the ground truth. Table 4.2 also presents the numbers of extraction patterns for the development set. For instance, a total of 17 patterns (i.e., combinations of spatial linguistic elements and distance elements) were developed to identify spatial indicators from the sentences. A total of 11 patterns (such as {T}{SI}{L}) were developed to extract SRTs from the sentences. The developed patterns together with the sentence structures were encoded as pattern-matching rules for information extraction.

Table 4.2. Number of patterns for sentence annotation and information extraction

Sentence annotation								
Sentence annotations	spatial entity	spatial entity modifier	spatial lexical unit	deontic operator indicator	negation indicator	distance value	distance unit	distance restriction
Number of annotation patterns	1(312) *	9	6(44) *	1	1(2) *	7	1(8) *	1(13) *
Information extraction								
Target information elements	Spatial indicator		Spatial relation triple		Trajector/Landmark attribute		Deontic operator indicator	Negation indicator
Number of extraction patterns	17		11		6		1	1

* Number in parenthesis represents the number of sub-concepts or gazetteer list elements

4.5.3 Evaluation, results, and analysis

The extraction performance was measured in terms of precision – the percentage of correctly extracted elements relative to the total number of elements extracted and recall – the percentage of correctly extracted elements relative to the total number of elements existing in the source text. The formalization performance was measured in terms of the accuracy in semantic formalization of the extracted elements – the percentage of correctly formalized elements relative

to the total number of correctly extracted elements and the accuracy in logic formalization of the HSLC 7-tuples – the percentage of correctly formalized sentences relative to the total number of positive sentences (i.e., those sentences whose contained 7-tuples were correctly extracted).

Table 4.3 presents the evaluation results for the test set. The ground truth includes 121, 23, 117, 136, 17, 86, and 27 elements of “Trajector”, “Trajector attribute”, “Spatial indicator”, “Landmark”, “Landmark attribute”, “Deontic operator indicator”, and “Negation indicator”, respectively, at a total of 527 elements. A performance of 94.7% recall and 98.2% precision was achieved in extracting information from the test set. A performance of 97.2% accuracy was achieved in formalizing the information elements. Regarding the DL formalization, 93.2% accuracy was achieved. The results demonstrate the effectiveness of the proposed approach in interpreting the spatial configuration-related requirements in utility regulations.

Table 4.3. Evaluation results for the test set

Number	Trajector	Trajector attribute	Spatial indicator	Landmark	Landmark attribute	Deontic operator indicator	Negation indicator	Total
Ground truth	121	23	117	136	17	86	27	527
Extracted	117	21	112	131	14	86	27	508
Correctly extracted	117	19	106	131	13	86	27	499
Correctly formalized	117	17	98	131	10	86	N/A	459
Extraction precision	100.0%	90.5%	94.6%	100.0%	92.9%	100.0%	100.0%	98.2%
Extraction recall	96.7%	82.6%	90.6%	96.3%	76.5%	100.0%	100.0%	94.7%
Formalization accuracy	100.0%	89.5%	92.5%	100.0%	76.9%	100.0%	N/A	97.2%

The following presents the findings through the analysis of the evaluation results.

First, 100% extraction precision of trajectors and landmarks indicates the effectiveness of UPO-based information extraction while the relatively low recall (96.7% and 96.3%, respectively) is attributed to the limited vocabulary size in UPO. In some cases, trajectors/landmarks are not explicitly prescribed in the sentences, which may also cause extraction errors. For example, in the following texts, the trajector “water line” and the landmark “ground”, both of which are not explicitly prescribed, were not extracted: “the horizontal separation between water and wastewater line” and “pipes buried underground”. Second, extraction of spatial indicators showed recall errors

(90.6%) mainly due to missing extraction patterns (which were not captured from the development set). For example, for the spatial indicator “come in contact with”, there is no matched pattern in the pattern set or matched spatial linguistic expression in SO. In some cases where prepositions such as “over”, “under”, “of” and “to” may not have spatial implications, spatial indicators that consist of these prepositions would also be falsely extracted. For example, in the text “urban streets where speed limits are under 30 mph”, “under”, which indicates a comparative relation to a speed value, was incorrectly extracted. There are also some uncommon spatial indicators that have degree adverbs such as “as near as practical to”. For these cases, only common parts of the spatial indicators such as “near” were extracted. As such, precision errors (94.6%) exist in extracting spatial indicators. Third, extraction of trajector/landmark attributes achieved the lowest recall and precision mainly because of missing extraction patterns and the GATE NLP tool (e.g., Stanford Parser) errors. There are two interesting cases of missing patterns. One is the pattern of trajectors/landmarks followed by prepositional phrases, and then followed by subordinate clauses, e.g., “in urban areas with curb and guttering where speed limits are 45 mph or greater, hydrants shall be placed 12 feet from the face of curb”, where the attribute “with curb and guttering” was extracted while the attribute of speed limit was not extracted. The other one is the pattern of independent subordinate clauses, e.g., “where speed limits are greater than 35 mph but less than 45 mph, hydrants shall be placed 8 feet from the face of curb”, where the attribute of speed limit was not extracted because there is no explicit spatial entity that the clause is to modify. No existing NLP tool achieves 100% performance. If multiple modifiers (e.g., adjective phrases, prepositional phrases, verb phrases, subordinate clauses) coexist for trajectors/landmarks, such as “all pipelines greater than 4 inches in outside diameter and crossing under non-controlled access highways carrying hazardous materials under pressure or having a wash factor”, it is challenging to generate the correct syntactic structure for distinguishing these modifiers, thus leading to missed extractions of attributes.

Regarding the performance in formalizing the extracted elements, 100% of trajectors and landmarks were correctly formalized, which is attributed to the success of UPO in capturing semantics in varying terms. The main errors exist in formalizing the trajector/landmark attributes (89.5% and 76.9%) and the spatial indicators (92.5%) due to the lack of their semantic correspondences in UPO/SO. For example, the attribute “public” of the landmark “right of way” was unformalized since there is no corresponding attribute modeled as the property of the

RightOfWay concept in UPO. Another example is the error of formalizing uncommon spatial indicators such as “placed normal to”, which do not have formal spatial relation mappings in SO. In certain cases, the extracted elements could be as complex as a subordinate clause, and thus, formalizing these elements is prone to errors. For example, for the attribute “where speed limits are 50 mph or greater”, its formalization was incorrect. In order to formalize these complex elements, additional work is required to analyze the elements, extract the key words, and map them to their corresponding ontology concepts/relationships. Regarding the DL-based formalization, the errors in formalizing the individual elements lead to the subsequent use of incorrect DL predicates/functions. In addition, there are some errors caused by not correctly specifying the negations (i.e., “ \neg ”) to certain DL functions. Thus, human efforts are also required to help interpret the negations to ensure the accurate formalization of the logic clauses.

4.6 Discussion

The newly developed NLP approach for the interpretation of utility regulations contributes to the body of knowledge in four aspects. First, the UPO is a deeper ontology that has an adequate vocabulary size and term diversity, which allows the extraction and formal representation of heterogeneous terminologies in regulations. Second, the SO contains two layers of semantics: linguistic spatial expressions and formal spatial relations, which enables the extraction of spatial semantics from natural language and advances existing NLP algorithms by incorporating spatial cognition. Third, extraction rules are encoded based on a set of text patterns that are formulated at the word, phrase, and sentence levels, which can reduce text ambiguities and enhance a deeper understanding of the requirements. Fourth, the mapping of extracted information elements to their semantic correspondences and transformation to logic clauses achieve the semantic and logic-based formalization of utility-specific regulatory knowledge, thus facilitating the objective and consistent interpretation of utility regulations. Jointly, the proposed NLP approach has the capability of spatial understanding and makes automatic spatial reasoning based on spatial information in texts feasible.

The NLP approach can be improved in three aspects for an even higher efficacy: 1) expand the analysis of regulatory requirements from the sentence level to document level; 2) enable the processing of quantitative requirements and existential requirements in addition to spatial requirements; and 3) develop methods to extract implicit regulatory knowledge (e.g., hidden

assumption and multiple exceptions), which is a challenge in the current method. In addition, DL was used in this study for the logical formalization of utility regulatory requirements despite the unavailability of an off-the-shelf deontic logic reasoner as its basic notations are fundamental for capturing the normative modalities of the requirements. In future work, the authors will extend existing FOL-based reasoners to add deontic reasoning capabilities to enable the direct DL reasoning (without further transformation into SPARQL) for utility compliance checking.

4.7 Summary and Conclusions

This paper presents an ontology and rule-based NLP approach to automate the interpretation of utility regulations. UPO and SO have been developed to facilitate the understanding of domain and spatial semantics. A set of text patterns have defined and encoded as pattern-matching rules for information extraction. A mechanism for information formalization has been designed towards the semantic and logic-based analysis of regulatory knowledge. The proposed approach was tested in extracting and formalizing spatial configuration-related requirements from utility accommodation policies. Results show the newly developed approach achieves 94.7% recall and 98.2% precision in information extraction and 93.2% accuracy in information formalization.

This study has the following conclusions. First, the ontology-based approach is effective in recognizing the domain technical and spatial terms from utility regulations as well as the semantic features attributed to the sufficiently large vocabulary size and the well-defined conceptualization in UPO and SO. Second, text patterns developed in this study well characterize the textual descriptions (e.g., words, phrases, and sentences) used in utility regulations, and thus, the pattern-matching rules are effective in extracting target information elements. Third, UPO and SO bridge the semantic gap between the natural language expressions and the formal semantics, and they are effective in guiding the formal representation of the extracted information; DL provides a logic-based format for representing the requirements; and therefore, the ontology and DL-based formalization enables the objective and consistent interpretation of textual regulatory requirements on underground utilities to ensure the compliance of underground utilities.

5. CONCLUSIONS

This chapter concludes the dissertation with a summary and discussions on the limitations and future studies.

5.1 Summary

Underground utilities must comply with the requirements stipulated in utility regulations to ensure their structural integrity and avoid interferences and disruptions of utility services. Noncompliance with the regulations could lead to utility incidents such as pipeline explosion and pipeline contamination, with disastrous consequences of property damages, environmental pollution, and personnel injuries and fatalities. Utility compliance checking is the action that examines the geospatial data of utilities and their surroundings against utility regulation data to identify the regulatory non-compliances in utility designs or existing records to limit possible negative impacts. However, the current practice of utility compliance checking mostly relies on manual efforts, which is time-consuming, costly, and error prone. This research offers an intelligent, knowledge-based method to automate the compliance checking of underground utilities.

In Chapter 2, this research first describes the development of an ontology-based framework for integrating heterogeneous geospatial and textual data of utilities and enabling automated compliance checking of underground utilities through semantic, logic, and spatial reasoning. The framework consists of the following key components: (1) four interlinked ontologies that provide the semantic schema for the representation of heterogeneous data relevant to utility compliance checking, (2) two data convertors for the conversion of heterogeneous data from proprietary formats into a common and interoperable format, and (3) a reasoning mechanism with spatial extensions for the detection of utility noncompliance. Under this framework, a more transparent implementation of utility compliance checking that are easy-to-understand and simple-to-implement even by non-experts is enabled, which is likely to shift this skill-based activity to a knowledge-based paradigm.

Next in Chapter 3, the research presents a novel method to develop a utility ontology that is semantically compatible with existing utility modeling initiatives and has a sufficient or

expandable vocabulary size to facilitate a high degree of interoperability across the utility infrastructure domain. The novel method integrates a top-down strategy and natural language processing (NLP) to develop the desired ontology from CityGML Utility Network ADE (a candidate open standard for modeling utility networks) and domain glossaries (lists of utility-specific terms and their textual definitions). This method contributes to increased levels of automation and efficiency in utility compliance checking by reducing laborious work on ontology development and offering a better option of interoperability facilitator for data integration.

In Chapter 4, the research presents the design of an ontology- and rule-based NLP approach to automate the interpretation of utility regulations – extracting the requirements from the regulations and further formalizing them into logic clauses – for supporting automated compliance checking of underground utilities. The approach integrates ontologies to capture both domain and spatial semantics in utility regulations and encode pattern-matching rules for information extraction. An ontology- and deontic logic-based mechanism is also integrated to facilitate the semantic and logic-based formalization of utility-specific regulatory knowledge. An end-to-end pipeline for automated interpretation of utility regulations is established, thus improving the level of automation in utility compliance checking by providing ready-to-use logic rules.

The methods and algorithms resulting from this research are tested using case studies and empirical experiments. The primary contribution of this research is the knowledge-based computational platform with semantic intelligence for regulatory compliance checking of underground utilities. The knowledge-based computational platform provides a declarative way rather than the otherwise procedural/hard-coding implementation approach to automate the overall process of utility compliance checking, which is expected to replace the conventional costly and time-consuming skill-based utility compliance checking practice. Deploying this computational platform for utility compliance checking will help eliminate non-compliant utility designs at the very early stage and identify non-compliances in existing utility records for timely correction, thus leading to enhanced safety and sustainability of the massive utility infrastructure in the U.S.

5.2 Limitations and Future Research

Throughout the journey of this research, several limitations have been identified, which are worth future research efforts. First, the ontology-based framework that is developed for integrating heterogeneous utility data for compliance checking has limited number of ontologies and data

convertors. In order to achieve a complete compliance checking through the utility asset life cycle, future research is needed to incorporate/develop ontologies and data convertors dedicated to other domains of knowledge in the utility industry, for example, utility construction, operation, and maintenance. Second, the semi-automated method for developing the utility ontology merely focuses on extracting the concepts that are relevant to utility physical products from semi-structured textual documents – domain glossaries. Future research is needed to extend the NLP approach to extract other types of concepts/relations from other types of textual documents. By incorporating new more semantics, the developed utility ontology can facilitate a higher degree of interoperability across the utility infrastructure domain. Hence, an increased level of efficiency of utility compliance checking by using the ontology can be achieved. Third, the NLP approach for interpreting utility regulations only focuses on the sentence-level processing of spatial configuration-related requirements in utility regulations. The approach can be improved in three aspects for an even higher efficacy: (1) expand the analysis of regulatory requirements from the sentence level to document level; (2) enable the processing of quantitative requirements and existential requirements in addition to spatial requirements; and (3) develop methods to extract implicit regulatory knowledge (e.g., hidden assumption and multiple exceptions), which is a challenge in the current method.

Last, one major limitation of this research is lack of an overall implementation that uses the ontology-based framework (as presented in Chapter 2), the developed utility ontology (as presented in Chapter 3), and the NLP-based interpretation of utility regulations (as presented in Chapter 4) for utility compliance checking. As an implementation prototype that is based on the ontology-based framework for utility compliance checking has been built and demonstrated as success (as presented in Chapter 2), future work will continue to incorporate the research outcomes from Chapters 3 and 4 into the prototype and use the same dataset to test the system for a higher level of semantics, automation, and efficiency in utility compliance checking, as illustrated in Figure 5.1. First, a domain ontology along with a sufficient and also expandable semantic dictionary (resulted from Chapter 3) establish a unified and uniform standard as the shared language for the utility domain. Supported by the semantic standard, all heterogeneous data such as geospatial data of utilities and textual data of regulations can be converted into the unified format of RDF (the RDF convertors resulted from Chapter 2 and the NLP approach resulted from Chapter 3 will be utilized), forming inter-connected RDF graphs. Further, advanced reasoning

algorithms can be designed to manipulate the RDF graphs via exploring, querying, and updating. For example, the mechanism that supports semantic, logic, and spatial reasoning as designed in Chapter 1 will be utilized. More importantly, the overall implementation of this research has big potential for practical contribution. The RDF graphs can accommodate advanced algorithms with adequate portability and interoperability including tools for connecting data sources, mapping and linking entities across digital objects, and integrating various features and applications over a heterogenous information network. As such, big data analytics that is embodied in semantic intelligence can be enabled within the RDF graphs for supporting a wide range of civil infrastructure applications. Utility compliance checking is such an application case.

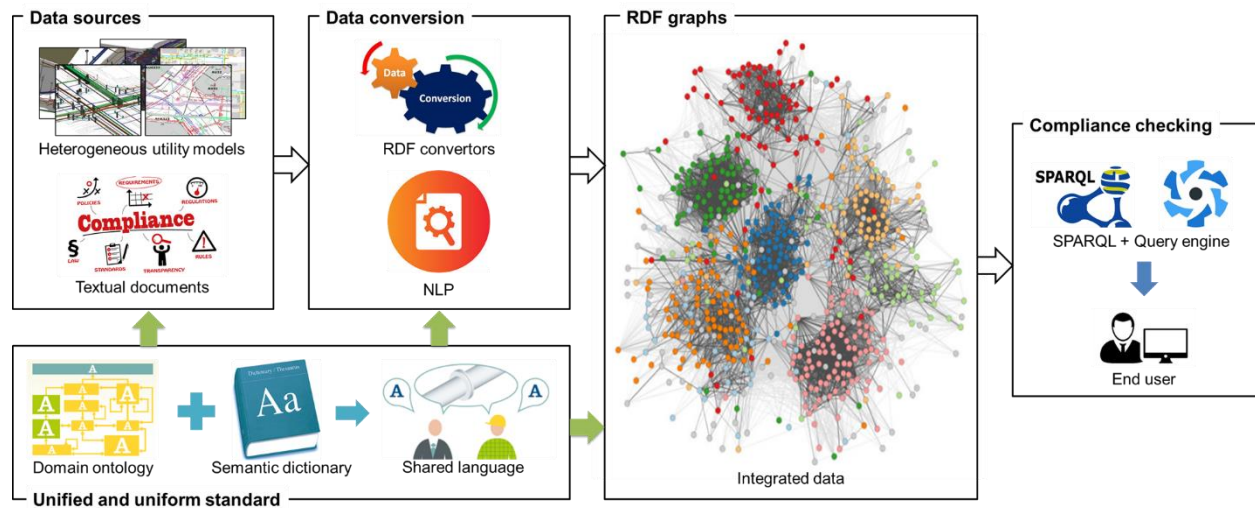


Figure 5.1. Future system implementation

One more research direction that is worth future efforts is to continuously explore NLP to enhance the human-machine interaction such as using natural/spoken language to interact with the computational platform of utility compliance checking, as shown in Figure 5.2, thus leading to advanced intelligence in utility infrastructure engineering and management.

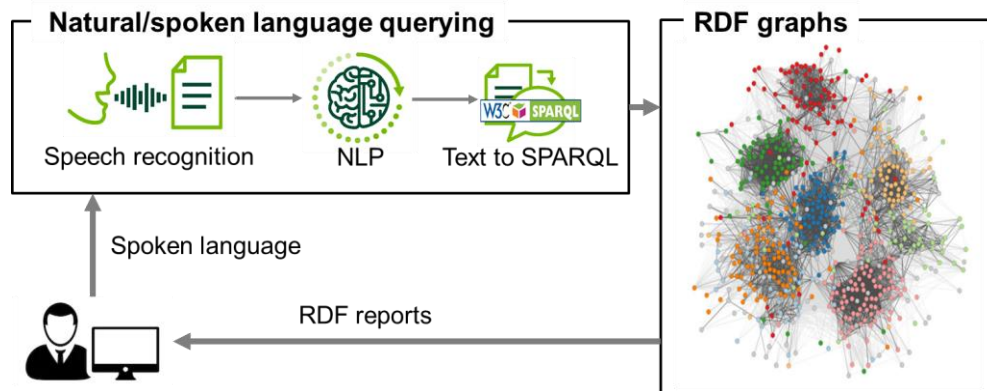


Figure 5.2. Spoken language-based human-machine interaction

REFERENCES

- [1] S. Li, H. Cai, V.R. Kamat, Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking, *Journal of Construction Engineering and Management*. 142 (2016) 04016074. doi:10.1061/(asce)co.1943-7862.0001199.
- [2] H.M. Osman, T.E. El-Diraby, Knowledge-Enabled Decision Support System for Routing Urban Utilities, *Journal of Construction Engineering and Management*. 137 (2010) 198–213. doi:10.1061/(asce)co.1943-7862.0000269.
- [3] National Transportation Safety Board, Pipeline accident report, Washington, DC, 2011.
- [4] National Transportation Safety Board, Pacific gas and electric company natural gas transmission pipeline rupture and fire San Bruno, California September 9, 2010, 2010. <http://www.nts.gov> (accessed May 9, 2019).
- [5] An Esri White Paper, ESRI Shapefile Technical Description, *Computational Statistics*. 16 (1998) 370–371. doi:10.1016/0167-9473(93)90138-J.
- [6] D. Abugov, N. Alexander, R. Anderson, B. Blackwell, R. Chatterjee, L. Angel Ramos Covarrubias, D. Geringer, M. Horhammer, Y. Hu, B. Kazar, R. Kothuri, S. Ravada, J. Wang, Q. Xie, J. Yang, Oracle® Spatial and Graph Developer's Guide 18c, 2018. <https://docs.oracle.com/en/database/oracle/oracle-database/18/spatl/spatial-and-graph-developers-guide.pdf> (accessed May 9, 2019).
- [7] Postgis, Postgis: Spatial and Geographic objects for PostgreSQL, <Http://Postgis.Net/>. (2013). <https://postgis.net/> (accessed May 9, 2019).
- [8] J. Steel, R. Drogemuller, B. Toth, Model interoperability in building information modelling, *Software and Systems Modeling*. 11 (2012) 99–109. doi:10.1007/s10270-010-0178-4.
- [9] OGC, CityGML | OGC, (2015). <https://www.opengeospatial.org/standards/citygml> (accessed May 9, 2019).
- [10] T. Le, H. David Jeong, Interlinking life-cycle data spaces to support decision making in highway asset management, *Automation in Construction*. 64 (2016) 54–64. doi:10.1016/j.autcon.2015.12.016.
- [11] P. Pauwels, S. Zhang, Y.C. Lee, Semantic web technologies in AEC industry: A literature overview, *Automation in Construction*. 73 (2017) 145–165. doi:10.1016/j.autcon.2016.10.003.

- [12] S.J. Fenves, E.H. Gaylord, S.K. Goel, Decision table formulation of the 1969 AISC specification, (1969) 1–167. <http://www.ideals.illinois.edu/bitstream/handle/2142/14275/SRS-347.pdf> (accessed May 9, 2019).
- [13] X. Tan, A. Hammad, P. Fazio, Automated Code Compliance Checking for Building Envelope Design, *Journal of Computing in Civil Engineering*. 24 (2010) 203–211. doi:10.1061/(asce)0887-3801(2010)24:2(203).
- [14] S. Malsane, J. Matthews, S. Lockley, P.E.D. Love, D. Greenwood, Development of an object model for automated compliance checking, *Automation in Construction*. 49 (2015) 51–58. doi:10.1016/j.autcon.2014.10.004.
- [15] J. Dimyadi, C. Clifton, M. Spearpoint, R. Amor, Regulatory Knowledge Encoding Guidelines for Automated Compliance Audit of Building Engineering Design, in: 2014: pp. 536–543. doi:10.1061/9780784413616.067.
- [16] O. Balaban, E. Sezen, Y. Kilimci, G. Cagdas, Automated Code Compliance Checking Model for Fire Egress Codes, *Digital Applications in Construction - ECAADe*. 2 (2012) 1–10.
- [17] J. Dimyadi, W. Solihin, W. Solihin, C. Eastman, A knowledge representation approach in BIM rule requirement analysis using the conceptual graph, *Journal of Information Technology in Construction*. 21 (2016) 370–402. <http://www.itcon.org/2016/24> (accessed October 23, 2019).
- [18] J. Qi, R.R.A. Issa, J. Hinze, S. Olbina, Integration of Safety in Design through the Use of Building Information Modeling, in: 2011: pp. 698–705. doi:10.1061/41182(416)86.
- [19] J. Choi, J. Choi, I. Kim, Development of BIM-based evacuation regulation checking system for high-rise and complex buildings, *Automation in Construction*. 46 (2014) 38–49. doi:10.1016/j.autcon.2013.12.005.
- [20] N.O. Nawari, Automating Codes Conformance, *Journal of Architectural Engineering*. 18 (2012) 315–323. doi:10.1061/(asce)ae.1943-5568.0000049.
- [21] F. Boukamp, B. Akinci, Automated processing of construction specifications to support inspection and quality control, *Automation in Construction*. 17 (2007) 90–106. doi:10.1016/j.autcon.2007.03.002.

- [22] C. Eastman, J. min Lee, Y. suk Jeong, J. kook Lee, Automatic rule-based checking of building designs, *Automation in Construction*. 18 (2009) 1011–1033. doi:10.1016/j.autcon.2009.07.002.
- [23] P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van De Walle, J. Van Campenhout, A semantic rule checking environment for building performance checking, *Automation in Construction*. 20 (2011) 506–518. doi:10.1016/j.autcon.2010.11.017.
- [24] T.H. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in the construction sector, *Expert Systems with Applications*. 42 (2015) 5219–5231. doi:10.1016/j.eswa.2015.02.029.
- [25] A. Costin, C. Eastman, Need for Interoperability to Enable Seamless Information Exchanges in Smart and Sustainable Urban Systems, *Journal of Computing in Civil Engineering*. 33 (2019) 04019008. doi:10.1061/(ASCE)CP.1943-5487.0000824.
- [26] S. Howell, Y. Rezgui, T. Beach, Integrating building and urban semantics to empower smart water solutions, *Automation in Construction*. 81 (2017) 434–448. doi:10.1016/j.autcon.2017.02.004.
- [27] J. Beetz, J. Van Leeuwen, B. De Vries, IfcOWL: A case of transforming EXPRESS schemas into ontologies, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*. 23 (2009) 89–101. doi:10.1017/S0890060409000122.
- [28] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Automation in Construction*. 63 (2016) 100–133. doi:10.1016/j.autcon.2015.12.003.
- [29] C. Métral, R. Billen, A.F. Cutting-Decelle, M. Van Ruymbeke, Ontology-based approaches for improving the interoperability between 3D urban models, *Electronic Journal of Information Technology in Construction*. 15 (2010) 169–184. <https://www.semanticscholar.org/paper/Ontology-based-approaches-for-improving-the-between-Métral-Billen/d68f60c02337600c237ae8a71946843185c0b9ed> (accessed May 9, 2019).

- [30] C. Metral, G. Falquet, A.F. Cutting-Decelle, Towards semantically enriched 3d city models : an ontology-based approach, in: *Proceeding GeoWeb 2009 Academic Track – Cityscapes - International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2009: pp. 40–45. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.46&rep=rep1&type=pdf> (accessed May 9, 2019).
- [31] K.H. Soon, R. Thompson, Semantics-based Fusion for CityGML and 3D LandXML, *3D Cadastre Workshop* 2014. (2014) 323–338. http://www.gdmc.nl/3dcadastres/literature/3Dcad_2014_33.pdf (accessed May 9, 2019).
- [32] A. Yurchyshyna, A. Zarli, An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction, *Automation in Construction*. 18 (2009) 1084–1098. doi:10.1016/j.autcon.2009.07.008.
- [33] B.T. Zhong, L.Y. Ding, H.B. Luo, Y. Zhou, Y.Z. Hu, H.M. Hu, Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking, *Automation in Construction*. 28 (2012) 58–70. doi:10.1016/j.autcon.2012.06.006.
- [34] T.E. El-Diraby, H. Osman, A domain ontology for construction concepts in urban infrastructure products, *Automation in Construction*. 20 (2011) 1120–1132. doi:10.1016/j.autcon.2011.04.014.
- [35] S. Howell, Y. Rezgui, T. Beach, Water utility decision support through the semantic web of things, *Environmental Modelling and Software*. 102 (2018) 94–114. doi:10.1016/j.envsoft.2018.01.006.
- [36] S.R. Mounce, C. Brewster, R.M. Ashley, L. Hurley, Knowledge management for more sustainable water systems, *Electronic Journal of Information Technology in Construction*. 15 (2010) 140–148.
- [37] T. Le, H. David Jeong, NLP-Based Approach to Semantic Classification of Heterogeneous Transportation Asset Data Terminology, *Journal of Computing in Civil Engineering*. 31 (2017) 04017057. doi:10.1061/(asce)cp.1943-5487.0000701.
- [38] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, *The Knowledge Engineering Review*. 11 (1996) 93–136. doi:10.1017/s0269888900007797.
- [39] N.F. Noy, D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, 2001. doi:10.1016/j.artmed.2004.01.014.

- [40] C. Cherpas, Natural language processing, pragmatics, and verbal behavior, *The Analysis of Verbal Behavior*. 10 (1992) 135–147. doi:10.1007/bf03392880.
- [41] N.W. Chi, Y.H. Jin, S.H. Hsieh, Developing base domain ontology from a reference collection to aid information retrieval, *Automation in Construction*. 100 (2019) 180–189. doi:10.1016/j.autcon.2019.01.001.
- [42] Y. Abuzir, “Moh’D Osama” Abuzir, Constructing the Civil Engineering Thesaurus (CET) Using ThesWB, in: *Computing in Civil Engineering* (2002), American Society of Civil Engineers, Reston, VA, 2004: pp. 400–412. doi:10.1061/40652(2003)34.
- [43] J. Zhang, N.M. El-Gohary, Extending Building Information Models Semi-Automatically Using Semantic Natural Language Processing Techniques, in: *Computing in Civil and Building Engineering* (2014), 2014: pp. 2246–2253. doi:10.1061/9780784413616.279.
- [44] Y. Rezgui, Text-based domain ontology building using Tf-Idf and metric clusters techniques, *Knowledge Engineering Review*. 22 (2007) 379–403. doi:10.1017/S0269888907001130.
- [45] X. Xu, H. Cai, K. Chen, Modeling 3D Spatial Constraints to Support Utility Compliance Checking, in: *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation - Selected Papers from the ASCE International Conference on Computing in Civil Engineering 2019*, American Society of Civil Engineers (ASCE), 2019: pp. 439–446. doi:10.1061/9780784482421.056.
- [46] S.K. Evt, S. Khayyal, V.E. Sanvido, Representing building product information using hypermedia, *Journal of Computing in Civil Engineering*. 6 (1992) 3–18. doi:10.1061/(ASCE)0887-3801(1992)6:1(3).
- [47] B. Feijó, W.G. Krause, D.L. Smith, P.J. Dowling, A hypertext model for steel design codes, *Journal of Constructional Steel Research*. 28 (1994) 167–186. doi:10.1016/0143-974X(94)90041-8.
- [48] E. Hjelseth, N. Nisbet, Capturing normative constraints by use of the semantic mark-up (RASE) methodology, in: *Proc., CIB W78-W102 Conf*, 2011: pp. 1–10. <http://itc.scix.net/data/works/att/w78-2011-Paper-45.pdf>.
- [49] J. Zhang, N.M. El-Gohary, Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking, *Journal of Computing in Civil Engineering*. 30 (2013) 04015014. doi:10.1061/(asce)cp.1943-5487.0000346.

- [50] P. Zhou, N. El-Gohary, Ontology-based automated information extraction from building energy conservation codes, *Automation in Construction*. 74 (2017) 103–117. doi:10.1016/j.autcon.2016.09.004.
- [51] J. Zhang, N.M. El-Gohary, Automated information transformation for automated regulatory compliance checking in construction, *Journal of Computing in Civil Engineering*. 29 (2015). doi:10.1061/(ASCE)CP.1943-5487.0000427.
- [52] Z. Li, K. Ramani, Ontology-based design information extraction and retrieval, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*. 21 (2007) 137–154. doi:10.1017/S0890060407070199.
- [53] E. Soysal, I. Cicekli, N. Baykal, Design and evaluation of an ontology based information extraction system for radiological reports, *Computers in Biology and Medicine*. 40 (2010) 900–911. doi:10.1016/j.combiomed.2010.10.002.
- [54] E. Arendarenko, T. Kakkonen, Ontology-based information and event extraction for business intelligence, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012: pp. 89–102. doi:10.1007/978-3-642-33185-5_10.
- [55] J. Tao, A. V. Deokar, O.F. El-Gayar, An ontology-based information extraction (OBIE) framework for analyzing initial public offering (IPO) prospectus, in: *Proceedings of the Annual Hawaii International Conference on System Sciences*, IEEE, 2014: pp. 769–778. doi:10.1109/HICSS.2014.103.
- [56] Major causes of Qingdao pipeline blasts identified- China.org.cn, (n.d.). http://www.china.org.cn/china/2014-01/09/content_31140285.htm (accessed May 9, 2019).
- [57] Thomas Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition*. 5 (1993) 199–220. <https://pdfs.semanticscholar.org/e790/2a46a83aa52ff8e2a36578a25f720fa648a2.pdf> (accessed May 9, 2019).
- [58] D.G. Sampson, M.D. Lytras, G. Wagner, P. Diaz, Ontologies and the Semantic Web for e-learning, *Educational Technology and Society*. 7 (2004) 26–28. doi:10.1007/978-3-540-92913-0_13.

- [59] W3C, RDF Schema 1.1, 2014. <https://www.w3.org/TR/rdf-schema/> (accessed May 9, 2019).
- [60] D. L. McGuinness, F. van Harmelen, OWL Web Ontology Language Overview, W3C Recommendation. (2004). <http://www.academia.edu/download/30759881/5.3-B1.pdf> (accessed May 9, 2019).
- [61] F. Manola, E. Miller, B. McBride, RDF 1.1 Primer, (2014). <https://www.w3.org/TR/rdf11-primer/> (accessed May 9, 2019).
- [62] M. Needleman, The unicode standard, *Serials Review*. 26 (2000) 51–54. doi:10.1080/00987913.2000.10764582.
- [63] R. Battle, D. Kolas, GeoSPARQL: Enabling a Geospatial Semantic Web, *Semantic Web Journal*. (2012) 1–17. doi:10.3233/SW-2012-0065.
- [64] C. Zhang, J. Beetz, B. De Vries, BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data, *Semantic Web*. 9 (2018) 829–855. doi:10.3233/SW-180297.
- [65] OGC, Geography Markup Language | OGC®, 5 (2009) 178–204. <http://www.opengeospatial.org/standards/gml> (accessed May 9, 2019).
- [66] OGC, CityGML | OGC, (2015). <http://www.opengeospatial.org/standards/citygml> (accessed May 9, 2019).
- [67] T.H. Kolbe, Representing and Exchanging 3D City Models with CityGML, in: *3D Geo-Information Sciences*, Springer Verlag, 2008: pp. 15–31. doi:10.1007/978-3-540-87395-2_2.
- [68] T. Becker, C. Nagel, T.H. Kolbe, Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization, in: *Lecture Notes in Geoinformation and Cartography*, 2013: pp. 41–62. doi:10.1007/978-3-642-29793-9-3.
- [69] I. Hijazi, M. Ehlers, S. Zlatanova, T. Becker, L. van Berlo, Initial Investigations for Modeling Interior Utilities Within 3D Geo Context: Transforming IFC-Interior Utility to CityGML/UtilityNetworkADE, in: 2011: pp. 95–113. doi:10.1007/978-3-642-12670-3_6.
- [70] Q.Z. Yang, Y. Zhang, Semantic interoperability in building design: Methods and tools, *CAD Computer Aided Design*. 38 (2006) 1099–1112. doi:10.1016/j.cad.2006.06.003.
- [71] F. Hogenboom, V. Milea, F. Frasincar, U. Kaymak, GeoSPARQL: A Geographic Query Language for RDF, *Emergent Web Intelligence Advanced Information Retrieval*. (2010) 87–116. <https://www.opengeospatial.org/standards/geosparql> (accessed May 9, 2019).

- [72] M. El-Mekawy, A. Östman, Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CITYGML, Proceedings of the 3rd ISDE Digital Earth Summit. (2010) 1–11. <http://www.diva-portal.org/smash/record.jsf?pid=diva2:392450> (accessed May 9, 2019).
- [73] A.E. Hor, G. Sohn, BIM-3DGIS Integrated Geospatial Information Model Using Semantic Web and RDF bipartite Graphs, ... -Remote-Sens-Spatial-Inf-Sci.Net. (2018) 2–3. doi:10.13140/RG.2.1.2816.7922.
- [74] W. Solihin, C. Eastman, Y.C. Lee, Multiple representation approach to achieve high-performance spatial queries of 3D BIM data using a relational database, Automation in Construction. 81 (2017) 369–388. doi:10.1016/j.autcon.2017.03.014.
- [75] J.K. Lee, C.M. Eastman, Y.C. Lee, Implementation of a BIM Domain-specific Language for the Building Environment Rule and Analysis, Journal of Intelligent and Robotic Systems: Theory and Applications. 79 (2015) 507–522. doi:10.1007/s10846-014-0117-7.
- [76] T.H. Beach, T. Kasim, H. Li, N. Nisbet, Y. Rezgui, Towards automated compliance checking in the construction industry, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Berlin, Heidelberg, 2013: pp. 366–380. doi:10.1007/978-3-642-40285-2_32.
- [77] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Automation in Construction. 53 (2015) 69–82. doi:10.1016/j.autcon.2015.03.003.
- [78] J. Dimyadi, R. Amor, Automating conventional compliance audit processes, in: IFIP Advances in Information and Communication Technology, Springer New York LLC, 2017: pp. 324–334. doi:10.1007/978-3-319-72905-3_29.
- [79] J. Dimyadi, G. Governatori, R. Amor, Evaluating LegalDocML and LegalRuleML as a Standard for Sharing Normative Information in the AEC/FM Domain, in: Researchspace.Auckland.Ac.Nz, 2017: pp. 637–644. doi:10.24928/jc3-2017/0012.
- [80] J. Zhang, N.M. El-Gohary, Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking, Automation in Construction. 73 (2017) 45–57. doi:10.1016/j.autcon.2016.08.027.

- [81] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean, SWRL: A Semantic Web Rule Language : Combining OWL and RuleML ; W3C Member Submission 21 May 2004, 2004. http://www.academia.edu/download/30680504/SWRL__A_Semantic_Web_Rule_Language_Combining_OWL_and_RuleM....pdf (accessed May 9, 2019).
- [82] M. Kifer, H. Boley, RIF Overview (Second Edition), W3C. (2013) 1–8. <https://www.w3.org/TR/rif-overview/> (accessed May 9, 2019).
- [83] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, J. Hendler, N3Logic: A Logical Framework For the World Wide Web, Cambridge.Org. (2007). <https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/n3logic-a-logical-framework-for-the-world-wide-web/5CB102B7E35457C8D07EC2B8281C8317> (accessed May 9, 2019).
- [84] C. Bizer, D2R Map: A Database to RDF Mapping Language, in: 12th World Wide Web Conference, 2003: pp. 2–3. <http://wwwconference.org/www2003/cdrom/papers/poster/p004/p4-bizer.html> (accessed May 9, 2019).
- [85] P. Pauwels, D. Van Deursen, J. de Roo, T. Van Ackere, R. de Meyer, R. Van de Walle, J. Van Campenhout, Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction, Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM. 25 (2011) 317–332. doi:10.1017/S0890060411000199.
- [86] N.M. El-Gohary, T.E. El-Diraby, Domain Ontology for Processes in Infrastructure and Construction, Journal of Construction Engineering and Management. 136 (2010) 730–744. doi:10.1061/(asce)co.1943-7862.0000178.
- [87] Washington State Water Reuse Workgroup, Pipeline Separation Design and Installation Reference Guide, 2006. <https://fortress.wa.gov/ecy/publications/publications/0610029.pdf> (accessed May 10, 2019).
- [88] J. Gundersen, S. Sarvis, Indot utility accommodation policy, (2013) 1–36. https://www.in.gov/indot/files/UC_UtilityAccommodationPolicy_061214.pdf (accessed May 10, 2019).

- [89] SPIN (SPARQL Inferencing Notation) | TopQuadrant, Inc, (n.d.). <https://www.topquadrant.com/technology/sparql-rules-spin/> (accessed May 9, 2019).
- [90] ESRI, ArcGIS Solutions, (2018). <https://solutions.arcgis.com/> (accessed May 9, 2019).
- [91] OpenStreetMap, OpenStreetMap, (2018). <https://www.openstreetmap.org/> (accessed May 9, 2019).
- [92] USGS National Elevation Dataset, (2013). <https://catalog.data.gov/dataset/usgs-national-elevation-dataset-ned> (accessed May 9, 2019).
- [93] M. Wang, Y. Deng, J. Won, J.C.P. Cheng, An integrated underground utility management and decision support based on BIM and GIS, *Automation in Construction*. 107 (2019). doi:10.1016/j.autcon.2019.102931.
- [94] K. Patroumpas, M. Alexakis, G. Giannopoulos, S. Athanasiou, TripleGeo: An ETL tool for transforming geospatial data into RDF triples, in: *CEUR Workshop Proceedings*, 2014: pp. 275–278. <http://www.dbnet.ece.ntua.gr/pubs/uploads/TR-2014-2.pdf> (accessed May 9, 2019).
- [95] X. Xu, H. Cai, Semantic approach to compliance checking of underground utilities, *Automation in Construction*. 109 (2020). doi:10.1016/j.autcon.2019.103006.
- [96] A.M. Costin, C. Eastman, R.R.A. Issa, The Need for Taxonomies in the Ontological Approach for Interoperability of Heterogeneous Information Models, in: *Congress on Computing in Civil Engineering, Proceedings, American Society of Civil Engineers (ASCE)*, 2017: pp. 9–17. doi:10.1061/9780784480830.002.
- [97] I. Boates, G. Agugiaro, A. Nichersu, Network modelling and semantic 3d city models: Testing the maturity of the utility network ADE for citygml with a water network test case, in: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018: pp. 13–20. doi:10.5194/isprs-annals-IV-4-13-2018.
- [98] T.A. Ei-Diraby, C. Lima, B. Feis, Domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge, *Journal of Computing in Civil Engineering*. 19 (2005) 394–406. doi:10.1061/(ASCE)0887-3801(2005)19:4(394).
- [99] T.E. El-Diraby, K.F. Kashif, Distributed ontology architecture for knowledge management in highway construction, *Journal of Construction Engineering and Management*. 131 (2005) 591–603. doi:10.1061/(ASCE)0733-9364(2005)131:5(591).

- [100] C.H. Caldas, L. Soibelman, J. Han, Automated classification of construction project documents, *Journal of Computing in Civil Engineering*. 16 (2002) 234–243. doi:10.1061/(ASCE)0887-3801(2002)16:4(234).
- [101] P. Zhou, N. El-Gohary, Domain-specific hierarchical text classification for supporting automated environmental compliance checking, *Journal of Computing in Civil Engineering*. 30 (2016). doi:10.1061/(ASCE)CP.1943-5487.0000513.
- [102] H. Fan, H. Li, Retrieving similar cases for alternative dispute resolution in construction accidents using text mining techniques, *Automation in Construction*. 34 (2013) 85–91. doi:10.1016/j.autcon.2012.10.014.
- [103] Y. Zou, A. Kiviniemi, S.W. Jones, Retrieving similar cases for construction project risk management using Natural Language Processing techniques, *Automation in Construction*. 80 (2017) 66–76. doi:10.1016/j.autcon.2017.04.003.
- [104] T. Kim, S. Chi, Accident Case Retrieval and Analyses: Using Natural Language Processing in the Construction Industry, *Journal of Construction Engineering and Management*. 145 (2019). doi:10.1061/(ASCE)CO.1943-7862.0001625.
- [105] M. Al Qady, A. Kandil, Concept Relation Extraction from Construction Documents Using Natural Language Processing, *Journal of Construction Engineering and Management*. 136 (2009) 294–302. doi:10.1061/(asce)co.1943-7862.0000131.
- [106] X. Jiang, A.H. Tan, Mining ontological knowledge from domain-specific text documents, in: *Proceedings - IEEE International Conference on Data Mining, ICDM, 2005*: pp. 665–668. doi:10.1109/ICDM.2005.97.
- [107] R. Navigli, P. Velardi, From glossaries to ontologies: Extracting semantic structure from textual definitions, *Frontiers in Artificial Intelligence and Applications*. 167 (2008) 71–87. [https://www.academia.edu/download/30772084/\(Buitelaar2008\)OntologyLearningandPopulationBridgingtheGapbetweenTextandKnowledge_292_.pdf#page=87](https://www.academia.edu/download/30772084/(Buitelaar2008)OntologyLearningandPopulationBridgingtheGapbetweenTextandKnowledge_292_.pdf#page=87) (accessed April 27, 2020).
- [108] S.H. Hsieh, H.T. Lin, N.W. Chi, K.W. Chou, K.Y. Lin, Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks, *Advanced Engineering Informatics*. 25 (2011) 288–296. doi:10.1016/j.aei.2010.08.004.

- [109] I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D.O. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals, in: ACL 2010 - SemEval 2010 - 5th International Workshop on Semantic Evaluation, Proceedings, 2010: pp. 33–38. https://dl.acm.org/ft_gateway.cfm?ftid=915830&id=1859670 (accessed April 27, 2020).
- [110] M. Baroni, R. Bernardi, N.Q. Do, C.C. Shan, Entailment above the word level in distributional semantics, in: EACL 2012 - 13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings, 2012: pp. 23–32. <https://dl.acm.org/citation.cfm?id=2380822> (accessed April 27, 2020).
- [111] N. Nakashole, G. Weikum, F. Suchanek, PATTY: A taxonomy of relational patterns with semantic types, in: EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference, Association for Computational Linguistics, 2012: pp. 1135–1145. www.mpi-inf.mpg.de/yago-naga/patty/ (accessed April 27, 2020).
- [112] R. Snow, D. Jurafsky, A.Y. Ng, Learning syntactic patterns for automatic hypernym discovery, in: Advances in Neural Information Processing Systems, 2005. <http://papers.nips.cc/paper/2659-learning-syntactic-patterns-for-automatic-hypernym-discovery.pdf> (accessed April 27, 2020).
- [113] Y. Rezgui, Text-based domain ontology building using Tf-Idf and metric clusters techniques, Knowledge Engineering Review. 22 (2007) 379–403. doi:10.1017/S0269888907001130.
- [114] D. Jurafsky, J.H. Martin, Speech and Language Processing 18 BT - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, in: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2009: p. 988. <http://nats-www.informatik.uni-hamburg.de/pub/CDG/JurafskyMartin00Comments/JurafskyMartin00-Review.pdf> (accessed June 22, 2019).
- [115] J. Pennington, R. Socher, C.D. Manning, GloVe: Global vectors for word representation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014: pp. 1532–1543. doi:10.3115/v1/d14-1162.

- [116] N.E. McTigue, J.M. Symons, The Water dictionary: a comprehensive reference of water terminology, American Water Works Association, 2010. doi:10.5860/choice.47-5394.
- [117] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, *Journal of Computing in Civil Engineering*. 24 (2010) 203–211. doi:10.1061/(ASCE)0887-3801(2010)24:2(203).
- [118] J.P. Martins, A. Monteiro, LicA: A BIM based automated code-checking application for water distribution systems, *Automation in Construction*. 29 (2013) 12–23. doi:10.1016/j.autcon.2012.08.008.
- [119] L. Jiang, R.M. Leicht, Automated rule-based constructability checking: Case study of formwork, *Journal of Management in Engineering*. 31 (2014). doi:10.1061/(ASCE)ME.1943-5479.0000304.
- [120] C. Preidel, A. Borrmann, J. Dimyadi, W. Solihin, Towards code compliance checking on the basis of a visual programming language, *Journal of Information Technology in Construction*. 21 (2016) 402–421. doi:http://www.itcon.org/2016/25. ISSN 1874-4753.
- [121] S. Park, J.-K. Lee, KBimCode-Based Applications for the Representation, Definition and Evaluation of Building Permit Rules, in: *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC)*, 2017. doi:10.22260/isarc2016/0087.
- [122] D.M. Salama, N.M. El-Gohary, Semantic Text Classification for Supporting Automated Compliance Checking in Construction, *Journal of Computing in Civil Engineering*. 30 (2016). doi:10.1061/(ASCE)CP.1943-5487.0000301.
- [123] M. Al Qady, A. Kandil, Automatic classification of project documents on the basis of text content, *Journal of Computing in Civil Engineering*. 29 (2015). doi:10.1061/(ASCE)CP.1943-5487.0000338.
- [124] W. Der Yu, J.Y. Hsu, Content-based text mining technique for retrieval of CAD documents, *Automation in Construction*. 31 (2013) 65–74. doi:10.1016/j.autcon.2012.11.037.
- [125] A.J.P. Tixier, M.R. Hallowell, B. Rajagopalan, D. Bowman, Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports, *Automation in Construction*. 62 (2016) 45–56. doi:10.1016/j.autcon.2015.11.001.

- [126] D.A. Salama, N.M. El-Gohary, Automated compliance checking of construction operation plans using a deontology for the construction domain, *Journal of Computing in Civil Engineering*. 27 (2013) 681–698. doi:10.1061/(ASCE)CP.1943-5487.0000298.
- [127] M.F. Moens, *Information extraction: Algorithms and prospects in a retrieval context*, Springer Netherlands, 2006. doi:10.1007/978-1-4020-4993-4.
- [128] P.J. Tierney, A qualitative analysis framework using natural language processing and graph theory, *International Review of Research in Open and Distance Learning*. 13 (2012) 173–189. doi:10.19173/irrodl.v13i5.1240.
- [129] J. Lee, J.S. Yi, J. Son, Development of Automatic-Extraction Model of Poisonous Clauses in International Construction Contracts Using Rule-Based NLP, *Journal of Computing in Civil Engineering*. 33 (2019) 04019003. doi:10.1061/(ASCE)CP.1943-5487.0000807.
- [130] K. Liu, N. El-Gohary, Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports, *Automation in Construction*. 81 (2017) 313–327. doi:10.1016/j.autcon.2017.02.003.
- [131] H. Cunningham, D. Maynard, K. Bontcheva, *Text Processing with GATE*, 2011. <https://dl.acm.org/citation.cfm?id=2018860> (accessed October 15, 2019).
- [132] X. Xu, H. Cai, K. Chen, An ontology approach to utility knowledge representation, in: *Construction Research Congress 2018: Infrastructure and Facility Management - Selected Papers from the Construction Research Congress 2018*, American Society of Civil Engineers (ASCE), 2018: pp. 311–321. doi:10.1061/9780784481295.032.
- [133] P. Kordjamshidi, M.F. Moens, Global machine learning for spatial ontology population, *Journal of Web Semantics*. 30 (2015) 3–21. doi:10.1016/j.websem.2014.06.001.
- [134] J. Cheng, Deontic relevant logic as the logical basis for representing and reasoning about legal knowledge in legal information Systems, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008: pp. 517–525. doi:10.1007/978-3-540-85565-1-64.
- [135] C. Prisacariu, G. Schneider, A formal language for electronic contracts, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Berlin, Heidelberg, 2007: pp. 174–189. doi:10.1007/978-3-540-72952-5_11.

- [136] G. Kołaczek, K. Juszczyszyn, Deontic logic-based framework for ontology alignment in agent communities, *Journal of Universal Computer Science*. 16 (2010) 178–197. doi:10.3217/jucs-016-01-0178.
- [137] U. Furbach, C. Schon, F. Stolzenburg, Automated reasoning in deontic logic, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 8875 (2014) 57–68. doi:10.1007/978-3-319-13365-2_6.
- [138] INDOT, INDOT Utility Accommodation Policy, (2013). https://www.in.gov/indot/files/UC_UtilityAccommodationPolicy_061214.pdf (accessed October 15, 2019).
- [139] GDOT, Utility Accommodation Policy and Standards, (2016). http://www.dot.ga.gov/PartnerSmart/utilities/Documents/2016_UAM.pdf (accessed October 15, 2019).