MACHINE LEARNING-BASED ANALYTICS OF STRUCTURED AND UNSTRUCTURED DATA FOR ENHANCED BRIDGE DETERIORATION PREDICTION

BY

KAIJIAN LIU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
with a concentration in Computational Science and Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Associate Professor Nora El-Gohary, Chair
Professor Khaled El-Rayes
Professor ChengXiang Zhai
Associate Professor Liang Y. Liu
Associate Professor Mani Golparvar-Fard

ABSTRACT

The increasing availability of heterogeneous bridge data from multiple sources opens unprecedented opportunities for data analytics to better predict bridge deterioration for supporting enhanced bridge maintenance decision making. Such data include structured National Bridge Inventory (NBI) and National Bridge Elements (NBE) data, structured traffic and weather data, and unstructured textual bridge inspection reports. However, despite the availability of the data, existing data-driven prediction methods mostly learn from abstract inventory data (e.g., the NBI data which describe bridge conditions by condition ratings) from a single source – missing the opportunity of leveraging the wealth of unstructured textual inspection reports and the diverseness of the multi-source data for enhanced deterioration prediction.

To capitalize on this opportunity, a novel bridge data analytics framework is proposed. The proposed framework is composed of six primary components: (1) a bridge deterioration knowledge ontology for facilitating semantic information and relation extraction from textual bridge inspection reports based on content and domain-specific meaning; (2) a semi-supervised machine learning-based semantic information extraction method for extracting information entities that describe bridge conditions and maintenance actions from the reports; (3) a supervised machine learning-based semantic relation extraction method for extracting dependency relations from the reports to link the extracted, yet isolated, information entities into concepts and to represent the semantically-low concepts in a semantically-rich structured way; (4) an unsupervised machine learning-based data linking method for linking the data records that are extracted from the reports and refer to the same entity; (5) a hybrid data fusion method for fusing the linked data records into a unified representation and for, subsequently, integrating the fused data with the other types of

structured data (i.e., NBI and NBE data, as well as traffic and weather data); and (6) a data-driven, deep learning-based bridge deterioration prediction method for learning from the integrated bridge data to predict the condition ratings of the primary bridge components and to predict the quantities of specific bridge element-level deficiencies.

The performance of the proposed framework was evaluated in predicting the deterioration of the state-owned bridges in Washington. It achieved a macro-precision and macro-recall of 89.9% and 85.8% when predicting the future condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures), and achieved a root mean square error, coefficient of variation, and coefficient of determination of 1.3, 27.6%, and 0.89, respectively, when predicting the future quantities of specific bridge element-level deficiencies. The experimental results demonstrated the promise of the proposed framework.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my profound gratitude to my advisor, Professor Nora El-Gohary, for her continuous guidance and unconditional support during my Ph.D. study. She is a responsible research advisor and a truly great life mentor. The positive influences she puts on me are tremendous and cannot be expressed enough in words. I am especially inspired by her immense dedication to the profession, her intense commitment to her work, and her wisdom of balancing work and life. I will forever remember and practice her inculcation: everything is about balance. I would also like to thank Professor ChengXiang Zhai for his support of my Computational Science and Engineering (CSE) Fellowship. I deeply appreciate the constructive advice and suggestions from the members of my Doctoral Committee – Professor Khaled El-Rayes, Professor ChengXiang Zhai, Professor Liang Y. Liu, and Professor Mani Golparvar-Fard.

I would like to give special thanks to my parents and younger sister for their love, encouragement, and support in my life. They are the very reason for all my achievements and that keeps me going. I would also like to thank my former groupmates, Prof. Jiansong Zhang, Prof. Lu Zhang, Prof. Xuan Lv, Prof. Peng Zhou, and Dr. Kadir Amasyali, and my current groupmates – Marwan Ammar, Lufan Wang, Ruichuan Zhang, Peter Liu, Nidia Bucarelli, and Xiyu Wang. It was my great pleasure to work with this group of talented people. I also thank all the rest of my friends and colleagues at the University of Illinois at Urbana-Champaign.

Finally, I gratefully acknowledge the funding support from the Strategic Research Initiatives (SRI) Program and the Computational Science and Engineering (CSE) Program by the Grainger College of Engineering, University of Illinois at Urbana-Champaign (UIUC), and the National Center for Supercomputing Applications (NCSA) at UIUC.

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

1.1 Introduction and Motivations

Bridges play an important role in ensuring the connectivity of transportation systems for providing daily mobility to the public. However, the U.S. bridges are in critical conditions and raise safety concerns. According to the American Society of Civil Engineers (ASCE)'s Infrastructure Report Card, the U.S. bridges received a grade of C+ (mediocre), with 9.1% and 13.6% of the nation's 614,387 bridges being structurally deficient and functionally obsolete, respectively (ASCE 2017). It is estimated that the average annual failure rate of the nation's bridges is between 87 and 222, with an expected value of 128 (Cook et al. 2013). Bridge failures are in some cases catastrophic and pose great threats to the safety of the public. For instance, the collapse of the I-35W Mississippi River Bridge – one of about 600 bridge failures that occurred in the U.S. between 1989 and 2013 – alone killed 13 people and injured 145 in 2007 (NTSB 2008). While bridge agencies are striving to improve the conditions of bridges, it is challenging to make cost-effective maintenance decisions under the stringent funding constraints. As estimated by the ASCE, in order to eliminate the nation's deficient bridge backlog by 2028, a $20.5 billion annual investment in the construction and maintenance of bridges is needed, while only $12.8 billion is being invested currently (ASCE 2013). Bridge maintenance decision making relies largely on the predicted future conditions of bridges and their elements to allocate the limited maintenance funding (Qiao et al. 2016; Zambon et al. 2017; Chang et al. 2019). With the increasing availability of data that can capture multiple factors related to the deterioration of bridges, there has been many demands for data-driven bridge deterioration prediction for supporting cost-effective maintenance decisions (FHWA 2016; NASEM 2016).

However, the current state-of-the-art data-driven bridge deterioration prediction methods/models are limited in this regard. On one hand, with the rapidly-evolving and expanding capabilities in data collection, large amounts of heterogeneous bridge data from multiple sources are becoming increasingly available. Such data include structured National Bridge Inventory (NBI) data, structured National Bridge Elements (NBE) data, and unstructured textual bridge inspection reports. In addition, structured traffic and weather data, which are relevant to bridge deterioration, are collected by responsible agencies such as the Federal Highway Administration (FHWA) and the National Oceanic and Atmospheric Administration (NOAA). On the other hand, despite the availability of the data, existing research efforts (e.g., Morcous 2011; Wellalage et al. 2014; Chang et al. 2017; Goyal et al. 2017; Lu et al. 2019) mostly focus on using abstract bridge inventory data from a single source – such as the NBI data which describe bridge conditions mainly by condition ratings – to predict, at a limited performance level, the future condition ratings of bridges. Such abstract data, although are very useful and important, are not sufficient, because they lack detailed descriptions about bridge conditions and maintenance actions, which limits the ability to learn from the history to predict the future deterioration. More specifically, existing data-driven methods/models are not capable of: (1) making use of the large amounts of rich data about bridge conditions and maintenance actions that are buried in textual inspection reports, which misses the opportunity of learning from such rich data for improved performance of bridge deterioration prediction (Washer et al. 2014); and (2) utilizing integrated data from multiple sources, which limits the capability to consider a diverse set of factors that may affect the deterioration of bridges (e.g., maintenance actions taken, material used in maintenance, traffic and weather patterns, etc.) and are, hence, important to consider when predicting the deterioration (Brown et al. 2014).

To address the aforementioned limitations, a novel bridge data analytics framework is proposed. The proposed framework is composed of six primary components, as per Figure 1.1, to allow for the extraction, integration, and analysis of both structured and unstructured data from multiple sources for enhanced bridge deterioration prediction. Three types of data are utilized in the proposed framework, including structured NBI and NBE data, structured traffic and weather data, and unstructured textual bridge inspection reports. Accordingly, the thesis research included seven primary research tasks: (1) conducting a comprehensive literature review; (2) developing a bridge deterioration knowledge ontology for facilitating semantic information and relation extraction from textual bridge inspection reports based on content and domain-specific meaning; (3) developing a semi-supervised machine learning (ML)-based semantic information extraction method and algorithm for extracting information entities that describe bridge conditions and maintenance actions from the reports; (4) developing a supervised ML-based semantic relation extraction method and algorithm for extracting dependency relations from the reports to link the extracted, yet isolated, information entities into concepts and to represent the semantically-low concepts in a semantically-rich structured way; (5) developing an unsupervised ML-based data linking method and algorithm for linking the data records that are extracted from the reports and refer to the same entity (e.g., the same type of deficiency on a bridge element); (6) developing a hybrid data fusion method and algorithm for fusing the linked data records into a unified representation and for, subsequently, integrating the fused data with the other types of structured data (i.e., NBI and NBE data, as well as traffic and weather data); and (7) developing a data-driven, deep learning-based bridge deterioration prediction method and algorithm for learning from the integrated bridge data to predict the condition ratings of the primary bridge components (i.e.,

decks, superstructures, and substructures) and to predict the quantities of specific bridge element-level deficiencies.

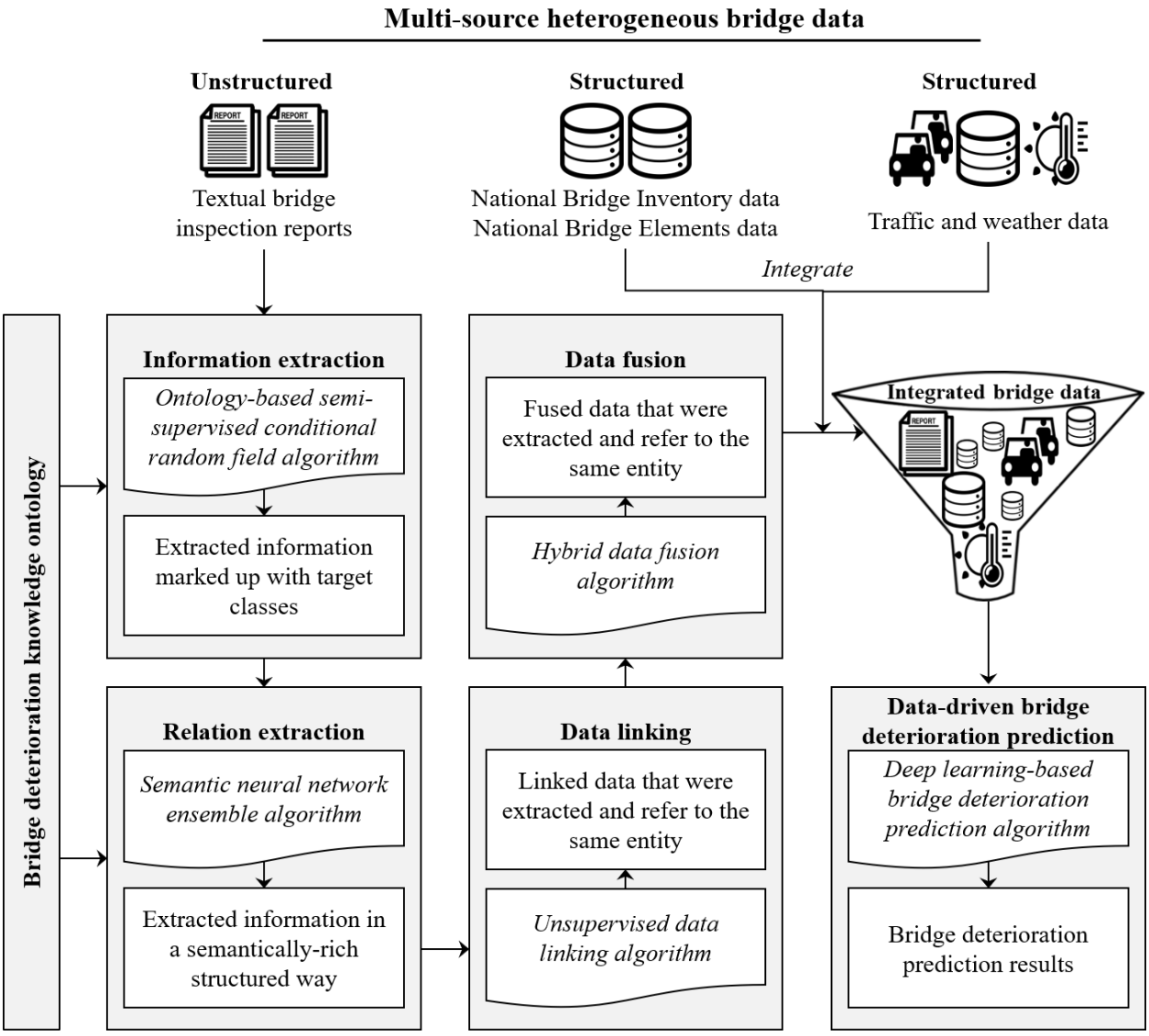


Figure 1.1. Proposed bridge data analytics framework.

1.2 State of the Art and Knowledge Gaps

1.2.1 State of the Art and Knowledge Gaps in Bridge Domain Ontologies

Ontology is defined as "an explicit specification of a conceptualization" for formally representing knowledge (Gruber 1995). An ontology that semantically represents bridge deterioration

knowledge is at the cornerstone of the proposed bridge data analytics framework. It aims to facilitate semantic information and relation extraction from textual bridge inspection reports based on content and domain-specific meaning. However, there is a lack of ontologies that sufficiently represent the knowledge of bridge deterioration for adequately supporting such text analytics. Accordingly, three primary knowledge gaps were identified.

First, existing ontologies provide a limited coverage of the core bridge deterioration knowledge aspects that are essential to analyze the semantics of the text to extract the needed information. For example, the ontology by El-Diraby and Kashif (2005) only represents the types of bridges without representing the deficiency types, which are essential to capture and extract information about how a bridge element has been or could be affected by different types of deficiencies. Similarly, the ontology by Bień et al. (2007) provides a limited coverage of bridge deficiencies and deficiency causes without any coverage of maintenance actions, which are essential to capture and extract information regarding how a deficiency has been or could be maintained.

Second, although existing ontologies can collectively cover the needed bridge deterioration knowledge aspects, they together still suffer from: (1) insufficient concept and/or relation coverage within each knowledge aspect. For example, the ontology by Kubota and Mikami (2013) only moderately covers the bridge element concepts, which limits the ontology's ability to support the extraction of bridge element concepts – especially when extracting agency-developed-elements that are defined and customized by different agencies to meet different bridge condition assessment needs; and (2) inconsistent and/or informal concept definition, conceptualization, and interpretation across the above-mentioned ontologies. For example, "slip" is defined as "the difference between the velocities of a solid surface and a fluid on the surface" by TRB (2015).

However, it is defined as “a deformation of the structure element caused by shear forces, without the deformation of the element cross-section” by Bień et al. (2007).

Third, existing ontologies are insufficient in capturing the classifications and multimodality views of bridge deterioration knowledge for semantically-rich information representation. For example, existing ontologies (e.g., Bien et al. 2007; TRB 2015) only model bridge deficiency concepts as a set of vocabularies without classifications or only with shallow classifications. Besides, existing ontologies (e.g., NCHRP 2011) only capture bridge maintenance action concepts as a list without modeling how these concepts could be classified according to different categorization criteria. Without in-depth and multi-view classifications, the extracted information from documents are semantically limited and may even mislead the representation, interpretation, and utilization of the extracted information. For instance, without a deeper classification, these two maintenance action concepts – “heat straightening” and “post-tensioning” – would be considered semantically equivalent, which could imply that these actions can be applied for a similar maintenance purpose; and, without multi-views or relationships, they cannot be represented according to the types of needed material, crew, cost, etc., and thus make the information lose important application contexts.

1.2.2 State of the Art and Knowledge Gaps in Information Extraction

Information extraction (IE), within this thesis, is defined as a named entity recognition and classification (NERC) task, which aims to automatically recognize and classify information entities into predefined entity classes. There is a body of research efforts – inside and outside of the civil engineering domain – that have been undertaken towards extracting information from unstructured text. Despite their achievements, existing IE methods are still limited in supporting

automated IE from complex, technical text – with highly-varying text patterns – such as that in bridge inspection reports. Accordingly, two primary knowledge gaps were identified.

First, there is a lack of IE methods that can simultaneously reduce human effort and achieve high performance when extracting information from highly heterogeneous and complex text. On one hand, most of the existing IE methods have taken a rule-based approach or a supervised ML-based approach. For example, almost all IE efforts in the construction domain have used rule-based IE methods (e.g., Abuzir and Abuzir 2002; Al Qady and Kandil 2010; Zhang and El-Gohary 2013; Zhou and El-Gohary 2015). Rule-based and supervised ML-based IE methods might be able to address the complexity and variability of text and thus achieve high IE performance by learning from a large set of representative examples, but they require a high amount of human effort. This is because such IE methods involve a human-intensive process for developing IE rules (in the case of rule-based IE) or annotating training examples (in the case of supervised ML-based IE). For example, the development of pattern-matching-based rules for the UMass MUC-4 system required 1,500 human-hours (Lehnert et al. 1991). It is even more challenging and more time-consuming to develop a comprehensive set of representative IE rules or annotations for text with highly-varying text patterns, such as that in bridge inspection reports. The utilization of incomprehensive and/or less representative rules or annotations could negatively affect the IE performance. One the other hand, although semi-supervised and unsupervised ML approaches offer plausible solutions to address this human-intensiveness problem, existing semi-supervised and unsupervised ML-based IE methods are still limited in extracting information from highly complex and variable text with high performance. They either followed a suboptimal algorithm for IE (e.g., Jiao et al. 2006; Mann and McCallum 2007; Liao and Veeramachaneni 2009; Liu et al. 2011) or did not explicitly capture the dependency structures of the natural language (e.g., Miller et al. 2004; Guo et al. 2009).

IE performance could be negatively affected by not explicitly representing and utilizing the dependency structures inherent in the natural language (Sutton and McCallum 2006).

Second, there is a lack of semantic ML-based IE methods. In recent years, a number of efforts explored the use of semantics for facilitating various natural language processing (NLP) tasks. For example, it was shown that the use of semantics – that are formally and explicitly defined by domain ontologies – improves the performance of domain-specific IE (e.g., Soysal et al. 2010; Zhang and El-Gohary 2013; Zhou and El-Gohary 2015). This is because formally defined semantics can assist in recognizing and extracting target information based on content and domain-specific meaning. Utilizing semantics for enhancing automated IE is therefore especially important for this research, given the complexity and variability of the text in bridge inspection reports. However, the utilization of formally defined semantics for supporting IE has been primarily studied in rule-based IE methods (e.g., Paassen et al. 2014; Zhang and El-Gohary 2013; Zhou and El-Gohary 2015). Most of the existing ML-based IE methods (e.g., Wu and Weld 2010; Qi et al. 2014) have only focused on representing text with syntactic features and/or less formally defined semantic features. The utilization of semantic features in ML-based IE differs from that in rule-based IE, because semantic features are meant to be interpreted by computers in the ML-based case rather than by human (when developing the rules). As such, the use of formally and explicitly defined semantics has not been well-explored in facilitating ML-based IE.

1.2.3 State of the Art and Knowledge Gaps in Relation Extraction

Relation extraction (RE), within this thesis, is defined as a dependency parsing (DP) task, which aims to automatically extract dependency relations between information entities in natural language text. There is a body of research efforts that have focused on developing ML-based DP models – using different learning techniques and various feature representations – for extracting

dependency relations from text. Despite the importance of these efforts, they cannot effectively extract dependency relations from highly technical, domain-specific text such as that in bridge inspection reports. Accordingly, three primary knowledge gaps were identified.

First, from an ML-based DP perspective, there is a lack of studies in ensemble learning-based DP methods. The majority of such methods (e.g., Yamada and Matsumoto 2003; Zhang and Clark 2008; Zhang and Nivre 2011; Chen and Manning 2014; Dyer et al. 2015; Cheng et al. 2016; Kiperwasser and Goldberg 2016; Hashimoto et al. 2017; Dozat and Manning 2017; Nguyen et al. 2017; Strubell and McCallum 2017; Dozat and Manning 2018) have focused on learning a single classifier to parse text for extracting dependency relations. Although a single classifier trained with advanced learning techniques (e.g., support vector machines and neural networks) could perform well on nonlinearly-separable instances/configurations, it is not sufficient to separate those with even more complex distributions (Sun et al. 2006; Bicke et al. 2007; Haixing et al. 2017) – such as the configurations of the text in the bridge reports (especially given that the reports have highly-varying levels of text characteristics and patterns). There are several efforts (e.g., Sagae and Lavie 2006; Nivre and McDonald 2008; Attardi and Dell'Orletta 2009; Hall et al. 2010) that proposed to integrate DP models at the parser level. For example, Nivre and McDonald (2008) proposed to integrate a graph-based parser and a transition-based parser by letting one parser generate features for the other one. Such methods are more co-training-based rather than ensemble learning-based. To the author's best knowledge, there is no ensemble learning-based DP method that utilizes a set of constituent classifiers to collectively capture the complex distributions of all the configurations for improved dependency relation extraction performance.

Second, from an ensemble learning perspective, there is a lack of studies in sampling training instances/configurations in a way that each constituent classifier is trained only with similarly-

distributed and thus more easily-separable configurations. Existing ensemble learning techniques (refer to Section 2.3.3) sample configurations based on simple, presumed distributions, such as the uniform distribution or weighted uniform distribution. Sampling configurations in this way cannot capture the configuration distribution characteristics of the text in bridge inspection reports, which makes it hard to generate meaningful configuration clusters and could thus make the trained constituent classifiers limited in collectively and sufficiently capturing the underlying distributions of all the configurations.

Third, from a feature representation perspective, there is a lack of studies that utilized semantic text features for facilitating DP. Existing DP methods (e.g., Bansal et al. 2014; Chen and Manning 2014; Guo et al. 2015) have relied on using distributed representations of syntactic features [e.g., words and part-of-speech (POS) features]. Although distributed representations could reveal the semantic meanings of the features to some extent, they provide limited semantics about word-to-word interactions that are important to consider when deciding on how sentences should be parsed. Such interactions can be better captured by the semantic features. For example, "maintenance material" and "maintenance action" are the semantic features for the words "concrete" and "patching", respectively. Based on the defined semantics – a maintenance material concept semantically describes a maintenance action concept – the dependency relation between "concrete", as a modifier word, and "patching", as a head word, could be correctly parsed and extracted.

1.2.4 State of the Art and Knowledge Gaps in Data Linking

Data linking, within this thesis, aims to link the data records that are extracted from bridge inspection reports and refer to the same entity. A number of research efforts have been undertaken towards developing data linking methods. Despite the importance of these efforts, they are still

limited in linking data extracted from highly technical, domain-specific documents, such as bridge inspection reports. Accordingly, three primary knowledge gaps were identified.

First, there is a lack of concept similarity assessment methods that are able to assess similarity in the absence of both contextual information and taxonomy-based concept mappings. To assess concept similarity, existing semantic similarity (SS) scoring functions either require the textual contexts of the concepts in a text corpus (e.g., Landauer 1998; Turney 2011), or need to map the concepts in comparison to their corresponding concepts in a taxonomy (e.g., Resnik 1995; Leacock and Chodorow 1998; Muller et al. 2006). Such prerequisite mapping is a challenge in itself, because it requires assessing the similarities between the concepts in the records and the concepts in the taxonomy. Zhang-El-Gohary similarity (Zhang and El-Gohary 2016) is the closest one that can address this challenge. It utilizes the WordNet taxonomy to calculate term-level SS scores and uses a scoring function to aggregate these scores into a concept-level SS score. As a result, only term-level mapping is needed. Term-level mapping, compared to concept-level mapping, is much more straightforward, because terms can be mapped by exact comparisons after stemming. However, this method generates asymmetrical similarities (i.e., the similarity of concept *x* to concept *y* is not equal to the similarity of *y* to *x*), because it compares a term of the first concept to all the terms of the second. Despite its success in its intended application, this method is not applicable to data linking in which symmetrical similarities are required (Christen 2012).

Second, there is a lack of record similarity assessment methods that can effectively assess the similarities of records when dependencies among attribute similarity assessments exist. These dependencies affect how record similarity should be assessed. For instance, in the following example, because the bridge element concepts in the two records (“floor beam splice” and “fascia stringer”) are already assessed as being different, there is no need to further assess the similarity

of the deficiency concepts ("flaking rust"): <floor beam splice, flaking rust> and <fascia stringer, flaking rust>. Existing data linking methods, especially clustering-based ones, mostly aggregate attribute similarities – either using equal (e.g., Elsner and Schudy 2000; Ng and Cardie 2002; Bilenko et al. 2005; Soon et al. 2006; Ailon et al. 2008; Elsner and Charniak 2008; Hassanzadeh et al. 2009) or different (e.g., Hassanzadeh et al. 2009; Hassanzadeh and Miller 2009; Haveliwala et al. 2009) attribute weights – into an overall similarity score for assessing record similarity, without taking such dependencies into account. It has been shown that such assessment methods are prone to generate a significant number of falsely-linked records (Ananthakrishna et al. 2002; Weis and Naumann 2004). On the other hand, a limited number of studies (e.g., Weis and Naumann 2004; Albrecht and Naumann 2008; Puhlmann et al. 2006) relied on general data schemas [e.g., Extensible Markup Language (XML) schema] to capture the structure of record attributes for assessing record similarity. However, such schemas cannot be used for capturing domain-specific dependencies such as those carried in the bridge report records.

Third, there is a lack of data linking methods that can address transitive closure problems. Existing linking methods, especially classification-based ones (e.g., Fellegi and Sunter 1969; Dey et al. 1998; Cochinwala et al. 2001; Elfeky et al. 2002; Bilenko and Mooney 2003; Christen 2008; Jiang et al. 2014), mostly follow the basic principle of the Fellegi-Sunter model. Such methods, thus, assume that the transitivity assumption holds: if (R_i, R_j) and (R_i, R_k) are linked respectively, then (R_j, R_k) is also linked, where R represents a record. These methods open the door to transitive closures, which typically leads to false positives (Elmagarmid et al. 2007; Christen 2012).

1.2.5 State of the Art and Knowledge Gaps in Data Fusion

Data fusion, within this thesis, aims to fuse the linked data records into a unified representation. The fusion requires two tasks. First, concept names that refer to the same entity, but vary in terms

of surface forms and abstraction levels, need to be fused into canonical identifier names. This is defined, within this thesis, as a named entity normalization task. Second, the numerical deficiency measures of the multiple instances, which are of the same type of deficiency but are at different locations of a bridge element, need to be fused into a single representative representation. This is defined, within this thesis, as a numerical data fusion task. A number of research efforts have been undertaken in the areas of named entity normalization and numerical data fusion. Despite the importance of these efforts, they are still limited in fusing data extracted from highly technical, domain-specific documents, such as bridge inspection reports. Accordingly, two primary knowledge gaps in each of the areas were identified.

In the area of named entity normalization, there is a lack of normalization methods that do not require human involvement in the normalization process. Most of the existing methods heavily rely on human-developed dictionaries or training data to normalize concept names (see Section 2.5.1). However, despite that several guidelines have defined the standard vocabularies used for structured bridge data (e.g., FHWA 1995; AASHTO 2010), there are no such guidelines for inspectors/writers – who have very different writing styles and specificity levels – to follow when choosing the concept names to use in the textual bridge inspection reports. As a result, the concept names used in the reports vary, to a high degree, in terms of surface forms and abstraction levels. It is challenging to develop/generate normalization dictionaries/data that can representatively and comprehensively capture such high-level variations. Second, there is a lack of normalization methods that are able to normalize concept names with both types of variations, such as those in bridge inspection reports. Most of the existing methods mainly focus on dealing with surface-form variations, which are caused by different naming conventions, e.g., acronyms and morphological variations. Yet, they are limited in normalizing concept names that also vary in terms of abstraction

levels (e.g., “north concrete bridge rail”, a subconcept of “bridge railing”). Balancing the abstraction and detailedness of the identifier names is critical to the ML-based bridge deterioration prediction model. As the features of the model, abstract identifiers (e.g., using “bridge” as the identifier of the aforementioned names) are too frequent in a collection of reports and, thus, lead to the loss of distinctive feature patterns. On the other hand, detailed identifiers (e.g., using “north concrete bridge rail”) are too rare in the collection and, thus, increase the dimensionality and the sparsity of the feature space, which would cause overfitting to a particular feature and therefore would undermine the generalizability of the model.

In the area of numerical data fusion, there is a lack of fusion methods that define the interval-based representation of the fused data in an objective way. Interval-based representations are usually used in major data fusion frameworks to characterize the uncertainty of the data (Sentz and Ferson 2002; Torra 2010). However, most of the existing methods (e.g., Zhang et al. 2017; Tian et al. 2018; He et al. 2018; Wu et al. 2018; Song et al. 2019) define the representation (i.e., defining the number of intervals and the size of the interval) in a subjective way. For example, based on subjective human judgement, Zhang et al. (2017) defined the representation of the fused building settlement data as four equal-size intervals. Subjective judgements are limited in defining the optimal number of intervals and the optimal size of the interval, because there is a tradeoff relationship between the two. A large number of intervals is preferred to capture more distinctive data instances for avoiding underfitting; and, at the same time, a large interval size is preferred to retain more data instances within an interval for avoiding overfitting. But, as the number increases, the size decreases. Such tradeoff is very difficult to balance using only subjective judgements of humans. Second, there is a lack of fusion methods that focus on fusing data that are complementary, such as the numerical deficiency measures in inspection reports, each of which

partially describes the overall condition of a deficiency. The majority of existing fusion methods (e.g., He et al. 2018; Zheng and Deng 2018; Xiao 2019; Mohammadi et al. 2019) focus on fusing data that are imprecise, conflicting, and/or multi-modal (Khaleghi et al. 2013), using the fuzzy set theory, Dempster-Shafer theory, and/or matrix factorization (Sentz and Ferson 2002; Lahat et al. 2015). When fusing complementary data (e.g., the deficiency measures), they would result in an interval-based representation that can only represent a subset of the data, which are less imprecise or conflicting but cannot fully capture the whole condition that the data collectively describe. Thus, despite being successful in their intended applications, existing data fusion methods are limited in fusing complementary data.

1.2.6 Sate of the Art and Knowledge Gaps in Data-Driven Bridge Deterioration Prediction

Data-driven bridge deterioration prediction, within this thesis, aims to learn from the integrated bridge data from multiple sources for predicting the future condition ratings of the primary bridge components and predicting the future quantities of specific bridge element-level deficiencies. A number of research efforts have been undertaken in the area of data-driven bridge deterioration prediction. Despite the importance of these efforts, they are still limited in supporting such a challenging prediction task. Accordingly, two primary knowledge gaps were identified.

First, there is a lack of methods that capitalize on the wealth of multi-source heterogeneous bridge data for enhanced deterioration prediction – that is not only able to predict the condition ratings of the primary bridge components with improved performance, but also able to predict the quantities of specific bridge element-level deficiencies. With the rapidly-evolving and expanding capabilities in data collection, large amounts of heterogeneous bridge data from multiple sources are becoming increasingly available, including structured NBI and NBE data, structured traffic and weather data,

and unstructured textual bridge inspection reports. Among them, previously-untapped textual inspection reports, which include a large amount of rich data/information describing bridge conditions and maintenance actions, are key data sources to allow for such enhanced prediction. However, despite the availability of such bridge data, existing data-driven prediction methods (e.g., Morcous 2011; Wellalage et al. 2014; Chang et al. 2017; Goyal et al. 2017; Lu et al. 2019) mostly focus on using abstract bridge inventory data from a single source – such as the NBI data which describe bridge conditions mainly by condition ratings – to predict the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures). Yet, due to mainly using abstract single-source data, their performance level is limited. Existing methods are, thus, limited in making use of the integrated bridge data that are originally in heterogeneous formats and from multiple sources – missing the opportunities of leveraging the wealth of textual inspection reports and the diverseness of the multi-source data for enhanced deterioration prediction.

Second, there is a lack of bridge deterioration prediction methods that are able to effectively learn from highly dimensional and imbalanced bridge data for supporting the prediction. Bridge data, especially integrated data from multiple sources, are of high dimensionality. For example, for a single bridge in the created dataset (refer to Section 8.2.2.1), its integrated data at a single timestep include 12,687 features, with 134, 1,480, 16, 196, and 10,861 features from the NBI data, NBE data, traffic data, weather data, and inspection report data, respectively. The high dimensionality of bridge data challenges the performance of data-driven methods in effectively predicting the deterioration. On the other hand, bridge data are naturally imbalanced; specifically, the numbers of bridges in different condition rating categories are imbalanced. For example, as of 2018, 2.5%, 14.8%, 42.0%, 24.7%, 12.3%, and 3.7% of the decks of the bridges in the U.S. are in the condition

rating categories of “excellent”, “very good”, “good”, “satisfactory”, “fair”, and “poor” or below, respectively. The imbalance in bridge data negatively affects the ability of data-driven methods to effectively capture the distribution characteristics of the data, which would undermine the performance of predicting the future condition ratings. However, existing data-driven prediction methods (e.g., Huang 2010; Creary and Fang 2015; Contreras-Nieto et al. 2016; Lim and Chi 2019) mostly leave these data challenges understudied or even untouched, which limits the ability to effectively learn from bridge data, which are highly dimensional and imbalanced.

1.3 Problem Statement

There is an emerging opportunity of leveraging machine learning-based data analytics to allow for the extraction, integration, and analysis of heterogeneous bridge data from multiple sources, in order to better predict bridge deterioration. However, there are two primary challenges to the utilization of multi-source heterogeneous bridge data: (1) heterogeneity: the data are structured and unstructured; and (2) complexity: the data are highly technical (i.e., having different levels of technical detail, text patterns, and text characteristics) and domain-specific, and are highly dimensional and imbalanced. There is no existing framework that is capable of dealing with the heterogeneity and complexity of the bridge data. In this regard, the following knowledge gaps were identified: (1) there is a lack of ontologies that sufficiently represent the knowledge of bridge deterioration for adequately supporting information and relation extraction from textual bridge inspection reports; (2) there is a lack of information extraction methods and algorithms that are able to effectively extract information that describes bridge conditions and maintenance actions from highly technical, domain-specific text, such as that in the textual reports; (3) there is a lack of relation extraction methods and algorithms that are able to effectively extract dependency relations from such text for representing the extracted information in a semantically-rich structured

way; (4) there is a lack of data linking methods and algorithms that are able to effectively assess the similarities between the data records extracted from the text and link the records without forming transitive closures; (5) there is a lack of data fusion methods and algorithms that are able to effectively fuse complex concept names (i.e., varying in terms of both surface forms and abstraction levels) and effectively fuse complementary numerical data in an objective way; and (6) there is a lack of data-driven bridge deterioration prediction methods and algorithms that are able to effectively learn from highly dimensional and imbalanced bridge data, which are originally in heterogeneous formats and from multiple sources, for better predicting the condition ratings of the primary bridge components and the quantities of specific bridge element-level deficiencies.

1.4 Research Objectives and Questions

The overall objective of the thesis research is to develop a bridge data analytics framework to allow for the extraction, integration, and analysis of multi-source heterogeneous (structured and unstructured) data for enhanced bridge deterioration prediction. Accordingly, six specific research objectives and outcomes were defined, along with the research questions.

(1) **Objective #1**: Develop a bridge deterioration knowledge ontology for facilitating semantic information and relation extraction from textual bridge inspection reports based on content and domain-specific meaning.

Research Questions: What are the concepts that need to be represented in the ontology to sufficiently cover the subject domain of knowledge (i.e., bridge deterioration knowledge) in terms of breadth, depth, classifications, and multimodality views? What are the concepts that need to be represented in the ontology to adequately support the subject application (i.e., semantic information and relation extraction from the reports)?

Outcome: An ontology that sufficiently represents the knowledge of bridge deterioration for adequately supporting semantic information and relation extraction from textual bridge inspection reports.

(2) **Objective #2**: Develop an ML-based semantic information extraction method for extracting information entities that describe bridge conditions and maintenance actions from textual bridge inspection reports.

Research Questions: What is the target information that needs to be extracted to capture the necessary information about bridge conditions and maintenance actions for supporting bridge deterioration prediction? What are the necessary features to represent the highly technical, domain-specific text in bridge inspection reports for information extraction? How to use the semantics of the ontology to conduct information extraction with high performance? How to develop information extraction algorithms that require as less human-annotation effort as possible, while achieving high performance (at least 85% in both precision and recall) – given the varied patterns and characteristics of the text?

Outcome: An ML-based semantic information extraction method and algorithm for automatically extracting information entities that describe bridge conditions and maintenance actions from textual bridge inspection reports.

(3) **Objective #3**: Develop an ML-based semantic relation extraction method for extracting dependency relations from textual bridge inspection reports for linking the extracted, yet isolated, information entities into concepts and representing the semantically-low concepts in a semantically-rich structured way.

Research Questions: What are the necessary features to capture the interrelationships between the information entities for relation extraction? How to use the semantics of the ontology to conduct relation extraction with high performance? How to develop relation extraction algorithms that are able to capture the complex distributions of all the configurations? How to develop relation extraction algorithms that are able to sample similarly-distributed and thus more easily-separable configurations into the same cluster? How to develop relation extraction algorithms that can achieve high performance (at least 80-85% in both precision and recall) when representing the extracted information in a semantically-rich structured way?

Outcome: An ML-based semantic relation extraction method and algorithm for automatically extracting dependency relations from the text for representing the extracted information in semantically-rich structured way.

(4) **Objective #4**: Develop an ML-based data linking method for linking the data records that are extracted from textual bridge inspection reports and refer to the same entity.

Research Questions: How to assess concept similarity in the absence of contextual information and taxonomy-based concept mappings? How to assess record similarity in the presence of dependencies among attribute (i.e., concept) similarity assessments? How to link the records without forming transitive closures for better linking performance? How to develop data linking algorithms that can automatically identify the optimal number of target clusters (i.e., the number of sets containing the linked records), without manually identifying this number?

Outcome: An ML-based data linking method and algorithm for linking the data records that are extracted from textual bridge inspection reports and refer to the same entity.

(5) **Objective #5**: Develop a hybrid data fusion method for fusing the linked data records into a unified representation and for, subsequently, integrating the fused data with the other types of structured data (i.e., NBI and NBE data, as well as traffic and weather data).

Research Questions: How to normalize the multiple concept names that refer to the entity, but vary in terms of both surface forms and abstraction levels, into a canonical name with balanced abstraction and detailedness? How to develop named entity normalization algorithms that do not require established lexicons in dictionaries and human-annotated training data? How to fuse numerical data that are complementary, such as the numerical deficiency measures in this research (i.e., each of the measures partially describes the overall condition of a deficiency)? How to define the interval-based representations of the fused data in an objective way? How to integrate the fused data with the other types of structured data (i.e., NBI and NBE data as well as traffic and weather data)?

Outcome: A hybrid data fusion method that includes three algorithms: (1) a named entity normalization algorithm for fusing the multiple concept names, (2) a numerical data fusion algorithm for fusing the multiple deficiency measures, and (3) a data integration algorithm for integrating the fused data with the other types of structured bridge data.

(6) **Objective #6**: Develop a data-driven, deep learning-based bridge deterioration prediction method for learning from the integrated bridge data (outcome of Objective #5) to predict the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures) and to predict the quantities of specific bridge element-level deficiencies.

Research Questions: How to develop prediction algorithms that are able to effectively learn from highly dimensional and sparse data, such as the integrated bridge data? How to develop

prediction algorithms that are able to effectively address the imbalance in data, such as the class imbalance in the integrated bridge data? How to capture the temporal dynamics that connect data over time for supporting the prediction? How to use deep learning techniques to support such a challenging data-driven prediction task?

Outcome: A data-driven, deep learning-based bridge deterioration prediction method and algorithm for predicting the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures) and for predicting the quantities of specific bridge element-deficiencies.

1.5 Research Tasks and Methodology

The research methodology includes seven primary research tasks, as summarized in Figure 1.2. A detailed introduction to each task and the corresponding research methodology is presented in the following subsections.

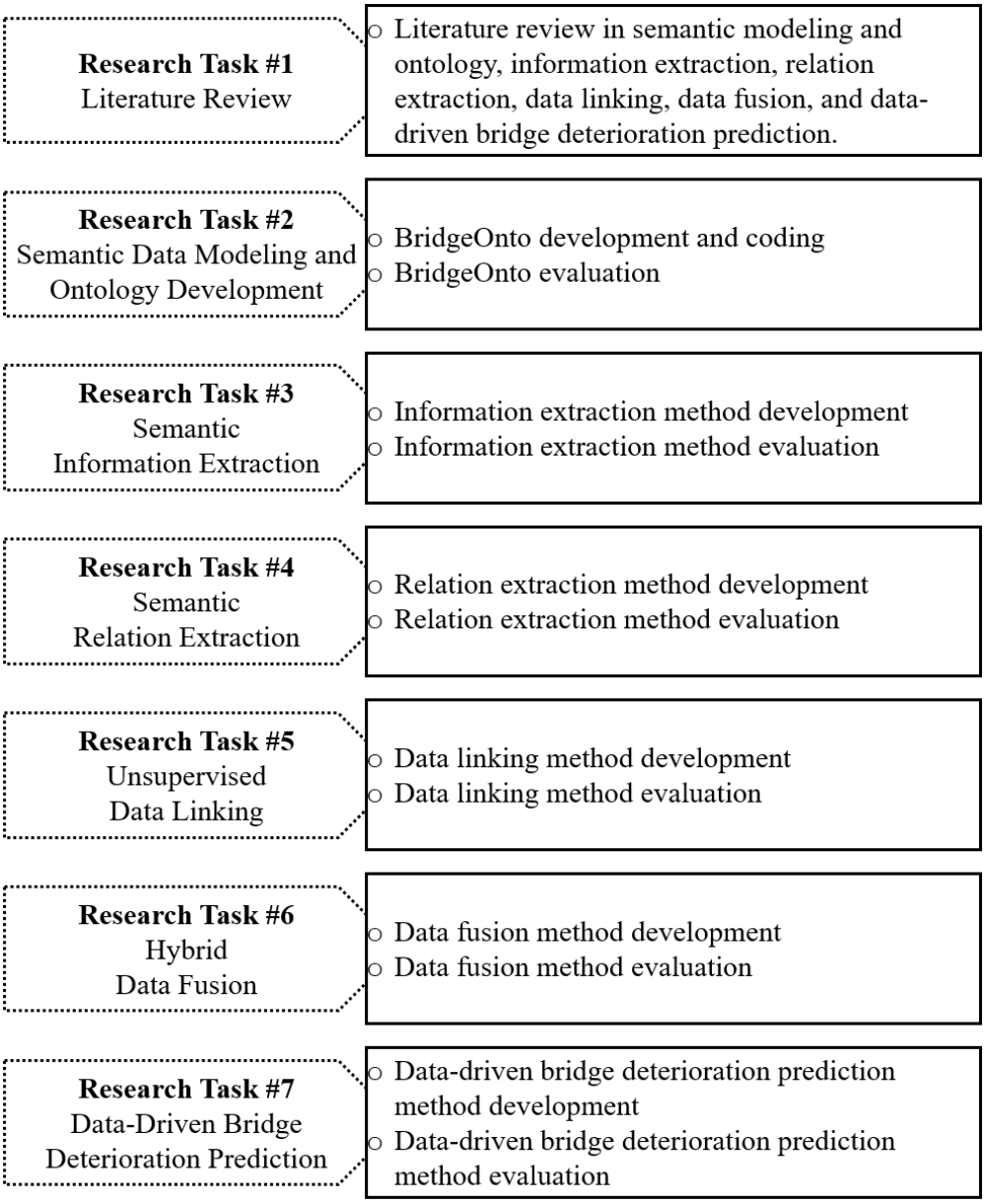


Figure 1.2. Research tasks and methodology.

1.5.1 Research Task #1 – Literature Review

The literature review covered six primary domains: semantic modeling and ontology, information extraction, relation extraction, data linking, data fusion, and data-driven bridge deterioration prediction. The following list summarizes the topics covered in each domain.

- Semantic modeling and ontology: the literature review focused on: (1) ontology development methodologies, and (2) existing semantic models and ontologies in the construction and civil infrastructure domain (with a focus on those related to bridge deterioration knowledge).
- Information extraction: the literature review focused on existing research and methods for information extraction in both the computer science and the construction and civil infrastructure domains. Specifically, the literature review covered: (1) existing rule-based information extraction methods and algorithms, (2) existing ML-based information extraction methods and algorithms (including supervised, semi-supervised, and unsupervised learning algorithms for information extraction), and (3) semantic similarity measures for assessing the similarities between terms/words (including corpus-based and knowledge-based semantic similarity measures).
- Relation extraction: the literature review focused on existing research and methods for relation extraction (i.e., dependency relation extraction, also known as dependency parsing) in both the computer science and the construction and civil infrastructure domains. Specifically, the literature review covered: (1) the transition-based dependency parsing model, (2) existing ML-based dependency parsing methods and algorithms (with a focus on neural network-based methods/algorithms, which are the current state of the art in the area of dependency parsing), and (3) existing ensemble machine learning methods.
- Data linking: the literature review focused on existing research and methods for data linking in both the computer science and the construction and civil infrastructure domains. Specifically, the literature review covered: (1) existing ML-based data linking methods and algorithms (including classification-based and clustering-based linking methods/algorithms),

(2) existing term similarity assessment functions, (3) existing concept similarity assessment functions, and (4) existing spectral clustering methods.

- Data fusion: the literature review focused on two areas: named entity normalization and numerical data fusion. In the area of named entity normalization, the literature review covered: (1) existing dictionary-based named entity normalization methods and algorithms, and (2) existing ML-based named entity normalization methods and algorithms. In the area of numerical data fusion, the literature review covered: (1) the commonly-used descriptive statistics in data fusion, and (2) existing data fusion theories and their applications.
- Data-driven bridge deterioration prediction: the literature review focused on existing research and methods in the areas of data-driven bridge deterioration prediction and machine learning (with a focus on deep learning). Specifically, the literature review covered: (1) existing data-driven bridge deterioration methods and algorithms (including deterministic, stochastic, and artificial intelligence-based prediction methods/models), (2) existing deep learning methods and algorithms (with a focus on recurrent neural networks), (3) existing manifold learning methods and algorithms, and (4) existing methods and algorithms for addressing data imbalance.

1.5.2 Research Task #2 – Semantic Data Modeling and Ontology Development

This research task aimed to develop a bridge deterioration knowledge ontology (namely, BridgeOnto) for facilitating semantic information and relation extraction from textual bridge inspection reports. This research task included two primary subtasks.

1.5.2.1 Subtask #2.1 – BridgeOnto Development and Coding

This subtask aimed to develop a domain-specific, unambiguous, and formalized representation of bridge deterioration knowledge, and to code it in Web Ontology Language (OWL) format. Benchmarking the ontology development methodology by El-Gohary and El-Diraby (2010), the development of the BridgeOnto included the following seven primary steps:

1. Domain, purpose, intended users, and scope definition: These fundamental scope descriptions were defined (as per Table 3.1) and utilized as guidance throughout the BridgeOnto development process.
2. Competency questions (CQs) development: A competency question (CQ) is expressed in the form of a natural language sentence that shows a pattern for a type of questions that an ontology must be able to answer (Fox and Gruninger 1998). CQs serve as functional requirements to ontologies. A set CQs for formulating the functional requirements to the BridgeOnto were developed and are discussed in Section 3.2.1.
3. Concept hierarchy construction: A concept hierarchy was constructed using two main iterative steps: (1) extracting key concepts from concept sources (the identified concept sources are explained in more detail in Section 3.1.2), and (2) organizing the extracted concepts into a concept hierarchy. The main concepts of the ontology were defined based on an analysis of a sample of bridge inspection reports, from both bridge engineering and NLP perspectives, in order to facilitate information and relation extraction from bridge inspection reports. At the highest level of abstraction, the BridgeOnto represents bridge deterioration knowledge by five main concepts: bridge element, deficiency, deficiency cause, maintenance action, and their related attributes (e.g., maintenance material, numerical measure, numerical measure unit, categorical quantity measure, categorical severity measure, and date). A combination of top-

down and bottom-up hierarchy construction approaches were used to avoid the inclusion of unnecessarily too detailed specific concepts and/or less-meaningful high-level concepts. The top-down approach first defines the most general concepts and then specifies their subconcepts; whereas, the bottom-up approach begins with defining the most specific concepts and then groups them into high-level concepts (Noy and McGuinness 2001).

4. Multimodality modeling: The concept hierarchy was reclassified based on different modality views for representing the polymorphic and multifaceted nature of bridge deterioration knowledge. Different modality views are shown Section 3.1.2.
5. Relation modeling: Three major types of relations were captured: (1) "is-a" relationship to characterize sub-superordinate relationships, (2) "is-part-of" relationship to decompose concepts into their constituent parts, and (3) cross-concept relationship to establish non-hierarchical relationships with semantic meanings between concepts.
6. Ontology capturing: Ontology capturing was conducted to define the formal terms of the concepts and relations.
7. Ontology coding: The BridgeOnto was coded using Protégé 3.4.5 (Protégé 2016). Protégé is an off-the-shelf ontology editor that supports coding ontology in OWL format. The coding included the following two main steps: (1) representing and coding the concepts as Protégé-OWL classes and using superclass-subclass relations to represent the hierarchical "is-a" relationships; and (2) representing and coding the relations using Protégé-OWL "extension property restrictions" and "necessary conditions".

1.5.2.2 Subtask #2.2 – BridgeOnto Evaluation

This subtask aimed to evaluate the developed ontology. The evaluation included verification and validation (Gómez-Pérez et al. 2006). Verification aimed to ensure that the ontology was

constructed correctly and consistently towards implementing the ontology requirements. The verification process included two main components: (1) answering CQs: verifying that the ontology meets its functional requirements, and (2) automated consistency and redundancy checking: verifying the freeness of the ontology from such errors. Validation aimed to evaluate the capability of the ontology in modeling the real-world that it tries to model. Validation is an important ontology quality assessment procedure that aims to assure the correctness of the knowledge encoded in an ontology (Vrandečić 2009). The ontology was validated using two techniques: (1) human expert validation: assessing how well the ontology meets the following criteria based on domain expert opinion: clarity, representation, coverage, conciseness, navigational ease, and extendibility; and (2) application-oriented validation: applying the ontology in a real-life application scenario (i.e., information and relation extraction from textual bridge inspection reports – as per Research Tasks #3 and #4) to evaluate its performance in its intended use.

1.5.3 Research Task #3 – Semantic Information Extraction

This research task aimed to develop an ML-based, semantic information extraction (IE) method and algorithm for extracting information entities that describe bridge conditions and maintenance actions from textual bridge inspection reports. This research task included two primary subtasks.

1.5.3.1 Subtask #3.1 – Information Extraction Method Development

IE, within this thesis, is defined as a named entity recognition and classification (NERC) problem. NERC aims to automatically recognize and classify information entities into predefined entity classes. As explained in Section 1.5.2.1, the entity classes were predefined based on the analyses of sample bridge inspection reports, from both bridge engineering and NLP perspectives. The

defined entity classes include: bridge element, deficiency, deficiency cause, maintenance action, maintenance material, numerical measure, numerical measure unit, categorical quantity measure, categorical severity measure, date, and other. This subtask focused on developing an ontology-based, semi-supervised conditional random fields (CRF)-based information extraction method and algorithm for extracting information entities of these entity classes from the inspection reports. It was composed of four main steps:

1. Baseline algorithm selection: In selecting the baseline algorithm(s), a number of existing rule-based and ML-based IE methods and algorithms were reviewed and analyzed (see Sections 1.2.2 and 1.5.1). Based on the review and analysis, the supervised CRF algorithm was selected as the baseline algorithm, because of its state-of-the-art IE/NERC performance. The baseline algorithm was also used to benchmark the performance of the proposed IE algorithm.
2. IE algorithm development: An ontology-based, semi-supervised CRF-based IE algorithm was developed. In developing the algorithm, a number of existing semi-supervised ML methods and algorithms were reviewed and analyzed (see Sections 1.2.2 and 1.5.1). Based on the review and analysis, the IE algorithm was developed under the semi-supervised learning cluster assumption: if two data points lay in the same cluster, they are likely to have a similar class label. This assumption was followed because it is the underlying assumption of most existing semi-supervised ML approaches.
3. Semantic feature representation: A semantic feature representation was developed to represent words in sentences for facilitating information extraction. In developing the semantic feature representation, existing feature representations for supporting IE were reviewed and analyzed (see Section 1.2.2). The semantic feature representation was then developed to include both syntactic features (i.e., lexical forms, stems, and POS tags of words) and semantic features

(i.e., semantic classes of words extracted based on the ontology developed as per Research Task #2). The semantic feature representation was used by the IE algorithm to learn how to extract information and to measure semantic similarities between labeled and unlabeled words.

4. Semantic similarity measurement: A semantic similarity (SS) measure was developed to derive the most likely entity class sequences for unlabeled data/sentences, so that the developed semi-supervised IE algorithm can learn from both labeled and unlabeled data. In developing the SS measure, existing semantic similarity measures were reviewed and analyzed (see Section 1.5.1). A heterogeneous information network meta-path-based SS measure was then developed, because it allows for capturing both corpus-based and knowledge-based semantic similarities between information entities (i.e., words). Capturing both types of similarities is essential for an accurate similarity measuring.

1.5.3.2 Subtask #3.2 – Information Extraction Method Evaluation

This subtask aimed to evaluate the performance of the developed IE methods and algorithms (both the proposed and the baseline). Evaluating the performance aimed to compare the algorithm-generated extraction results against the gold standard using evaluation metrics. Precision and recall were selected as the primary evaluation metrics. Precision is the percentage of the total number of correctly-extracted information entities out of the total number of all extracted entities. Recall is the percentage of the total number of correctly-extracted entities out of the total number of entities that should be extracted. F-1 measure, as the weighted harmonic mean of recall and precision, was also selected. Because the proposed IE method deals with a multi-class classification problem where each information entity could be labeled with one of the eleven defined entity classes, average precision, recall, and F-1 measure were also used as evaluation metrics, which are the arithmetic means of precisions, recalls, and F-1 measures over all the entity classes. The details of

method implementation, including dataset preparation, are presented in Section 4.2.2. The evaluation results are presented and discussed in Section 4.3.

1.5.4 Research Task #4 – Semantic Relation Extraction

This research task aimed to develop an ML-based, semantic relation extraction (RE) method and algorithm for extracting dependency relations from textual bridge inspection reports to link the extracted, yet isolated, information entities into concepts and to represent the semantically-low concepts in a semantically-rich structured way. This research task included two primary subtasks.

1.5.4.1 Subtask #4.1 – Relation Extraction Method Development

RE, within this thesis, is defined as a dependency parsing (DP) problem. DP aims to recognize and extract word-to-word dependency relations from the text for linking words (i.e., information entities extracted as per Research Task #3) into concepts and for representing the semantically-low concepts in a semantically-rich structured way. This subtask focused on developing a semantic neural network ensemble (NNE)-based DP method and algorithm for automatically extracting dependency relations from the text. It was composed of four main steps:

1. Baseline algorithm selection: In selecting the baseline algorithms, existing rule-based and ML-based DP methods and algorithms were reviewed and analyzed (see Sections 1.2.3 and 1.5.1). Based on the review and analysis, three DP algorithms were selected as the baselines for benchmarking and evaluating the performance of the proposed algorithm, including semantic single classifier-based algorithms that use a single neural network (NN) or a single support vector machine (SVM) classifier and a semantic stacked generalization-based algorithm that use cross-validation partitioning for sampling the configurations.

2. DP algorithm development: In developing the DP algorithm, existing ensemble learning and dependency parsing methods and algorithms were reviewed and analyzed (see Sections 1.2.3 and 1.5.1). Based on the review and analysis, NN-based and SVM-based DP algorithms were selected as the bases for developing the proposed semantic NNE-based DP algorithm. The NN algorithm was used for developing constituent classifiers for the proposed algorithm. The SVM algorithm was used for developing a combiner classifier for the proposed algorithm.
3. Semantic distributed feature representation: In developing the semantic distributed feature representation, existing feature representations for supporting DP were reviewed and analyzed (see Section 1.2.3). A new semantic distributed feature representation, which uses configuration-based features, syntactic and semantic text features, and distributed feature representation, was then developed for representing the configurations. Configurations in the semantic distributed feature representations were used by the DP algorithms (the proposed and the baseline) to learn how to extract dependency relations.
4. Similarity-based sampling: In developing the sampling algorithm, the characteristics of the configuration distributions were analyzed, and existing ensemble learning methods and algorithms were reviewed and analyzed (see Sections 1.2.3 and 1.5.1). A similarity-based sampling algorithm was then developed to sample configurations into configuration clusters (defined based on the characteristics of the configuration distributions and are further explained in Section 5.2.1.2) in a way that a cluster only contains similarly-distributed and thus more easily-separable configurations. The similarities between the configurations and the transition centers were used to sample configurations into the clusters.

1.5.4.2 Subtask #4.2 – Relation Extraction Method Evaluation

This subtask aimed to evaluate the performance of the developed relation extraction/DP methods and algorithms (both the proposed and the baseline). The evaluation included algorithm validation and testing. Algorithm validation was conducted, using the configurations, to: (1) select the hyperparameter values for the classifiers, (2) select the feature representation, and (3) compare the performance of the proposed DP algorithm to those of the three baselines. The selection and comparison were conducted based on configuration-based accuracy, which is the ratio of the number of correctly-classified configurations to the total number of configurations. Algorithm testing was conducted, using the testing sentences, to evaluate the performance of the proposed DP algorithm (with the selected hyperparameters and feature representation) in extracting dependency relations from bridge inspection reports for representing the extracted information in a semantically-rich structured way. The performance was measured in terms of precision, recall, and F-1 measure, at both the semantic information element (SIE) and semantic information set (SIS) levels. Precision is the ratio of the number of correctly-extracted SIEs/SISs to the total number of extracted SIEs/SISs. Recall is the ratio of the number of correctly-extracted SIEs/SISs to the total number of SIEs/SISs that should be extracted. F-1 measure is the weighted harmonic mean of precision and recall. A threefold cross-validation was performed to evaluate the generalizability of the algorithm. The confidence intervals of the mean values for these measures were also calculated to evaluate the sensitivity of the performance results. These evaluation metrics were calculated by comparing the algorithm-predicted extractions with the gold standard annotations. The details of method implementation, including dataset preparation, are presented in Section 5.2.2. The evaluation results are presented and discussed in Section 5.3.

1.5.5 Research Task #5 – Unsupervised Data Linking

This research task aimed to develop an ML-based data linking method and algorithm for linking data records that are extracted from textual bridge inspection reports and refer to the same entity. This research task included two primary subtasks.

1.5.5.1 Subtask #5.1 – Data Linking Method Development

Data linking, within this thesis, aims to link the data records that are extracted from the reports and refer to the same entity. For example, the following two records were extracted from the same bridge inspection report (LaDOTD 2008) and refer to the same entity (i.e., the crack on the girder web): <box girders webs, crack, several> and <longitudinal steel box girder webs, cracks, many>. This subtask focused on developing a spectral clustering (SC)-based data linking method and algorithm for linking the data records. It was composed of four main steps:

1. Development of concept similarity assessment method: A new concept similarity (CS) assessment method was developed, which assesses the similarities between concepts based on the similarity degrees of their terms, without the need for pre-existing context information or taxonomy-based concept mappings. In developing the method, three alternative CS scoring functions were developed and tested. The most suitable function was selected based on the testing results.
2. Development of record similarity assessment method: A new sequential record similarity assessment method was developed, which breaks down the record-level similarity assessment task into sequences of attribute-level tasks based on similarity assessment dependencies. Similarity assessment dependencies indicate that: (1) the record similarity assessment should be conducted as sequences of attribute similarity assessment tasks, where the similarities of

object concepts should be assessed prior to assessing the similarities of property concepts; and (2) the similarities of object concepts decide if there is a need to further assess the similarities of property concepts. For example, because the bridge element (object) concepts in the two records are already assessed as being different, there is no need to further assess the similarity of the deficiency (property) concepts: <floor beam splice, flaking rust> and <fascia stringer, flaking rust>. Accordingly, three types of similarity assessment dependencies for the bridge report record similarity assessment were defined: element-deficiency, element-deficiency cause, and element-maintenance action. The method was evaluated based on its effectiveness in supporting the linking.

3. Selection of data linking method: SC was selected for linking the records at each attribute level for five main reasons. First, data linking can be naturally formulated as a graph-partitioning task, where same/similar records (as vertices) are partitioned into the same subgraph and are thus linked (as edges). Second, SC does not make a strong assumption on the shapes of target clusters (Long et al. 2006; Zhang et al. 2008). This is much desired because records do not necessarily lie in disjoint convex sets. Third, it embeds high-dimensional data into a linear, low-dimensional space by representing the $n \times n$ Laplacian matrix using an $n \times k$ matrix (where $k \ll n$). The resulting matrix contains only a few leading eigenvectors of the Laplacian matrix (Chan et al. 1994; Doyle et al. 2008), and thus avoids the curse of dimensionality (Doyle et al. 2008). Because of the linearity, the clustering results are always at global maxima. Fourth, previous studies have shown that it outperforms "traditional" clustering methods, such as k-means and single linkage (Long et al. 2006; Zhang et al. 2008). Fifth, it can be easily implemented, because its eigen-decomposition process can be solved efficiently by standard linear algebra methods (Zhang et al. 2008; Lei and Rinaldo 2015).

4. Development of improved SC-based data linking method: An improved SC-based data linking method was proposed, which uses iterative bi-partitioning to automatically identify the optimal number of target classes (the number of sets containing linked records). The original SC method requires manually defining this number (Meila 2016), which is challenging because the number varies across datasets (e.g., across different bridge inspection reports) and the true number for each dataset is unknown (if without human-annotated gold standards). In addition to this improvement, the use of unsupervised pre-classification prior to the clustering – to break down a similarity graph into several small ones – was tested to evaluate if the size reduction of the graph would improve the clustering performance. Both, the pre-classification and the iterative bi-partitioning, were evaluated based on their effectiveness in supporting the linking.

1.5.5.2 Subtask #5.2 – Data Linking Method Evaluation

This subtask aimed to evaluate the performance of the developed data linking methods and algorithms (both the proposed and its variations). The linking results were compared to those in the gold standard, and were evaluated based on example-based precision, recall, and F-1 measure. Using the example-based measures, the data linking performance was calculated for each record in a report, and the overall performance was obtained by calculating the mean performance over all the records in the report. The example-based precision is the average of the ratio of the number of correctly-linked records to the total number of linked records across all the records extracted in a report. The example-based recall is the average of the ratio of the number of correctly-linked records to the total number of records that should be linked across all the records extracted in a report. The example-based F-1 measure is the weighted harmonic mean of the example-based precision and recall. The details of method implementation, including dataset preparation, are

presented in Sections 6.2.2 and 6.3. The evaluation results are presented and discussed in Section 6.3.

1.5.6 Research Task #6 – Hybrid Data Fusion

This research task aimed to develop a hybrid data fusion method and algorithm for fusing the linked data records into a unified representation and for, subsequently, integrating the fused data with the other types of structured data (i.e., NBI and NBE data, as well as traffic and weather data). This research task included two primary subtasks.

1.5.6.1 Subtask #6.1 – Data Fusion Method Development

Data fusion, within this thesis, aims to fuse the linked data records (extracted from bridge inspection reports as per Research Tasks #3 and #4, and linked as per Research Task #5) into a unified representation and to integrate the fused data with the other types of structured data. This subtask focused on developing hybrid data fusion method and algorithms. The developed method includes three algorithms: a named entity normalization (NEN) algorithm for fusing concept names, a numerical data fusion algorithm for fusing numerical deficiency measures, and a data integration algorithm for integrating the fused report data with the other types of structured data (i.e., NBI and NBE data as well as traffic and weather data). It was composed of three main steps:

1. NEN algorithm development: In developing the NEN algorithm, existing NEN methods and algorithms were reviewed and analyzed (see Sections 1.2.5 and 1.5.1). An unsupervised NEN algorithm was then developed to include a concept ranking function and a concept selection rule for normalizing concept names. The ranking function considers the corpus statistic score, term-position score, and term-sequence score of a candidate identifier concept name to calculate its ranking score. Ranking functions with different combinations of the three types of

scores were also developed and tested. The most suitable function was selected based on the testing results. The selection rule considers both the corpus statistics and the lexical patterns of concept names to select a final candidate identifier name from the top-raking names. Different combinations of two hyperparameters, which are used by the rule to balance the abstraction and detailedness of the identifier concept names, were tested. The combination with the optimal hyperparameter values was selected based on the testing results.

2. Numerical data fusion algorithm development: In developing the fusion algorithm, existing numerical data fusion theories, methods, and algorithms were reviewed and analyzed (see Sections 1.2.5 and 1.5.1). Based on the review and analysis, the fusion algorithm was then developed to use interval-based representations for representing the fused data, because they can account for the uncertainty in data and can avoid the exaggerated impact of minor fluctuations in continuous data on the machine learning-based prediction models. It was developed to use information entropy as the main fusion criterion for fusing the data which are complementary, because information entropy can quantify how well an interval-based representation can represent such data.
3. Data integration algorithm development: In developing the integration algorithm, the characteristics of the bridge data (i.e., NBI, NBE, traffic, weather, and fused report data) were analyzed, and the main integration criteria were then identified. Based on the analysis and the identified integration criteria, the integration algorithm was then developed to integrate the fused report data with the structured NBI and NBE data based on the structure identification number, and to, subsequently, integrate these data with structured traffic and weather data based on the spatial distances between bridges and traffic/weather monitoring stations. The detailed implementation of the integration is further explained in Section 8.2.2.2.

1.5.6.2 Subtask #6.2 – Data Fusion Method Evaluation

This subtask aimed to evaluate the performance of the developed data fusion methods and algorithms (both the proposed and its variations). Only the normalization and fusion algorithms need to be evaluated. The integration algorithm does not require evaluation, because the integration is a straightforward and error-free process. The evaluation included method verification and validation. Method verification aimed to evaluate the correctness of the fusion method. The NEN algorithms (the proposed and its variations) were verified based on accuracy, which is the number of correct identifier concept names out of the total number of identifier concept names. The developed entropy-based fusion algorithm was verified based on information entropy, which is equal to zero if the algorithm can stably fuse the same set of data instances into the same interval in a simulation run; otherwise, it increases from zero. Method validation aimed to evaluate the performance of the fusion method in supporting its intended use – fusing data extracted from bridge inspection reports for supporting bridge deterioration prediction (i.e., predicting the future condition ratings of decks, superstructures, and substructures). Two main types of prediction models were developed: using fused data and using unfused data. The performances of the prediction models developed using the fused data were compared to the performances of the models developed using the unfused data, in order to evaluate the performance of the fusion method. The performance results were compared based on average accuracy, which is the average of the ratio of the number of correctly-predicted condition ratings to the total number of ratings per condition rating category. The details of method implementation, including dataset preparation, are presented in Section 7.2.2. The evaluation results are presented and discussed in Section 7.3.

1.5.7 Research Task #7 – Data-Driven Bridge Deterioration Prediction

This research task aimed to develop a data-driven, deep learning-based bridge deterioration prediction method and algorithm for learning from integrated bridge data to predict the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures) and to predict the quantities of specific bridge element-level deficiencies. This research task included two primary subtasks.

1.5.7.1 Subtask #7.1 – Data-Driven Bridge Deterioration Prediction Method Development

This subtask focused on developing a data-driven, deep learning-based prediction method and algorithm that is able to learn from the integrated bridge data, which are highly dimensional and imbalanced, for predicting the condition ratings of bridges and the quantities of specific bridge element-level deficiencies. It was composed of four main steps:

1. Selecting and extending the method for dealing with data dimensionality: In selecting the method for extension, existing manifold learning (also known as dimensionality reduction) methods and algorithms were reviewed and analyzed (see Sections 1.2.6 and 1.5.1). Based on the review and analysis, the isometric feature mapping (Isomap) algorithm (Tenenbaum et al. 2000) was selected for embedding the high-dimensional and sparse bridge data into a low-dimensional dense space. Because the Isomap algorithm requires assessing the distances between data instances (which, in this research, include both numerical and categorical features), a revised Euclidean distance was proposed to allow for the distance assessment of the data instances with the mixed types of features.
2. Recurrent neural network (RNN) development: An RNN architecture was developed to learn from the embedded bridge data from past years to predict the conditions of bridges and their

elements in the next year. In developing the architecture, three main criteria were followed: (1) the ability of the architecture to capture the temporal dynamics that connect data over time, (2) the ability of the architecture to capture the dimensionality and the nonlinearity of data, and (3) the computational efficiency of the architecture. An RNN architecture, which includes an input layer, a recurrent layer, a pooling layer, a set of nonlinear dense layers, and an output layer, was then developed.

3. Selecting and extending the method for dealing with data imbalance: In selecting the method for extension, existing data sampling and cost-sensitive learning methods and algorithms were reviewed and analyzed (see Sections 1.2.6 and 1.5.1). Based on the review and analysis, the cost-sensitive learning approach was selected, because it does not increase or decrease the size of a dataset, which helps avoid overfitting and the loss of important data instances. The binary focal loss function, which is used for conducting cost-sensitive learning, was selected because it uses a modulating factor to directly adjust the learning cost. Since predicting the condition ratings of bridges is a multi-class classification problem, the binary focal loss function was extended into a multi-class focal loss function.
4. Bridge deterioration prediction algorithm development: A deep learning-based bridge deterioration prediction algorithm was developed. It combines the Isomap algorithm, the RNN architecture, and the multi-class focal loss function to predict bridge deterioration. Two baseline algorithms were also developed to benchmark the performance of the proposed algorithm in predicting the condition ratings. The first baseline learned from the integrated multi-source bridge data, using the RNN architecture, but with the cross-entropy loss function. Unlike the multi-class focal loss function used in the proposed algorithm, the cross-entropy loss function treats the cost of misclassifications in the minority classes and the cost of

misclassifications in the majority classes equally and, hence, does not address the imbalance in the data. The second baseline is same as the first (i.e., used the RNN architecture with the cross-entropy loss), but only learned from the NBI data. The second baseline algorithm is similar to existing data-driven bridge deterioration prediction methods/algorithms, which mostly focus on learning from single-source bridge inventory data (e.g., NBI data or similar inventory data collected by different countries), without addressing data imbalance. Only the proposed algorithm was used for predicting the quantities of deficiencies for two main reasons. First, to the author's best knowledge, there is no existing data-driven prediction method/algorithm that is able to predict the detailed quantity of a specific bridge element-level deficiency, which provides no benchmark for direct comparison. Second, learning from NBI data solely is not applicable in this case, since they do not include such detailed data about bridge element-level deficiencies.

1.5.7.2 Subtask #7.2 – Data-Driven Bridge Deterioration Prediction Method Evaluation

This subtask aimed to evaluate the performance of the developed bridge deterioration prediction methods and algorithms (both the proposed and the baseline). The following two metrics were used for evaluating the performance of predicting the condition ratings: macro-precision and macro-recall. Macro-precision and macro-recall measure the overall performance using the mean of the precision and recall for each condition rating category, respectively. Precision is the ratio of the number of correctly-predicted condition ratings to the total number of predicted ratings for a category. Recall is the ratio of the number of correctly-predicted condition ratings to the total number of ratings that should be predicted for a category. The following three metrics were used for evaluating the performance of predicting the deficiency quantities: root mean square error (RMSE), coefficient of variation (CV), and coefficient of determination (R^2). RMSE measures, on

average, how concentrated the predicted data are around the line that best fits the actual data. CV measures the extent to which the overall prediction error varies with respect to the mean of the actual data. R^2 measures the percentage of the variance of the actual data explained by the prediction model. The details of method implementation, including dataset preparation, are presented in Section 8.2.2. The evaluation results are presented and discussed in Section 8.3.

1.6 Contribution to the Body of Knowledge

1.6.1 Intellectual Merit

This thesis research offers a novel bridge data analytics framework to allow for the extraction, integration, and analysis of both structured and unstructured data from multiple sources for enhanced bridge deterioration prediction. It contributes to the body of knowledge in seven primary ways.

- First, this research offers a new bridge deterioration knowledge ontology. The ontology advances the knowledge modeling efforts in the bridge domain by sufficiently capturing the bridge deterioration knowledge (about bridge element, deficiency, deficiency cause, maintenance action, and their related attributes) in terms of breadth, depth, classifications, and multimodality views. The ontology has shown effectiveness in adequately supporting semantic information and relation extraction from bridge inspection reports and, hence, is expected to be able to support similar text analytics tasks in the bridge domain.
- Second, this research offers a new ontology-based, semi-supervised conditional random fields-based information extraction method for extracting information that describes bridge conditions and maintenance actions from bridge inspection reports. The method offers a way for semantically and simultaneously capturing the dependency structures as well as the

distributions of a small set of fixed labeled data and a large set of unlabeled data in a semi-supervised yet concave objective function for machine learning. Its capability of dynamically adapting itself to unseen instances by further learning from the unlabeled data and its concavity nature allow the needed information extraction to be conducted effectively and in an efficient way that requires less human effort.

- Third, this research offers a new semantic neural network ensemble-based relation extraction method for extracting dependency relations from bridge inspection reports to represent the unstructured text in a semantically-rich structured way. The method offers a new way for ensemble learning, which allows each of the multiple constituent neural network classifiers to only learn from similarly-distributed and thus more easily-separable data instances (sampled by the proposed similarity-based sampling method), in order to better capture the complex distributions of all the data instances collectively for supporting more effective ensemble learning. This new ensemble learning approach, compared to the traditional approaches that use simple, presumed distributions for data sampling, allows the extraction of dependency relations from highly technical, domain-specific text (such as that in textual inspection reports) to be conducted more effectively.
- Fourth, this research offers a new unsupervised data linking method for linking data records that are extracted from the reports and refer to the same entity. The method leverages improved spectral clustering to analyze the similarities between data instances for effectively linking data in a completely unsupervised manner, without human involvement. It offers new knowledge on how to assess concept similarity in the absence of both contextual information and taxonomy-based concept mappings, how to assess record similarity in the presence of dependencies among attribute similarity assessments, how to automatically identify the

optimal number of target class without using a manually identified number, and how to conduct data linking in an unsupervised way without forming transitive closures.

- Fifth, this research offers a new hybrid data fusion method for fusing the linked data extracted from bridge inspection reports into a unified representation. The named entity normalization algorithm of the method uses corpus statistics and lexical patterns to fuse concept names, which offers new knowledge on how to fuse complex concept names that vary in terms of both surface forms and abstraction levels into identifier concept names that balance the abstraction and detailedness, without human involvement. The numerical data fusion algorithm of the method uses data discretization and information entropy to fuse numerical deficiency measures into a single representative representation, which offers new knowledge on how to fuse complementary data in an objective way.
- Sixth, this research offers a new deep learning-based prediction method for learning from integrated bridge data from multiple sources for enhanced bridge deterioration prediction. The proposed method uses a number of machine (deep) learning techniques to support such a challenging prediction task, including deep learning, manifold learning, and cost-sensitive learning. It, thus, offers new knowledge on how to effectively learn from data that are highly dimensional and imbalanced for better predicting bridge deterioration.
- Seventh, this research offers a novel bridge data analytics framework, which allows for using multi-source heterogeneous data for enhanced bridge deterioration prediction – that is not only able to predict the condition ratings of bridges with improved performance, but also able to predict the quantities of specific bridge element-level deficiencies. On one hand, this research goes beyond the current state of the art in data analytics, where data in heterogeneous formats (i.e., structured and unstructured) are mostly analyzed separately. On the other hand, it goes

beyond the current state of the art in data-driven bridge deterioration prediction, where existing methods mostly use abstract bridge inventory data to predict – at a limited performance level – the condition ratings of bridges.

More detailed discussions of the intellectual merit of each of the aforementioned methods and contribution to the body of knowledge are provided in Chapter 9.

1.6.2 Broader Impacts

The research outcomes could bring the following significant benefits to the society at large:

- Promoting the use of unstructured textual data in the bridge domain: Unstructured textual bridge data, such as bridge inspection reports, include a large amount of detailed information describing bridge conditions and maintenance actions – much beyond what can be found in structured bridge data, such as the NBI data. Yet, due the challenges in analyzing textual data, the wealth of unstructured textual bridge data is not being fully harnessed. Using the proposed information and relation extraction methods, data users in the bridge domain (e.g., maintenance decision makers) could gain improved access to the rich data/information buried in these unstructured data sources. One important benefit of using textual bridge data, which has been demonstrated in this research, lies in extracting information from bridge inspection reports and using it in machine learning for improved performance of bridge deterioration prediction. Many more benefits can be expected when these methods are applied to other types of textual data for supporting data-driven applications in the bridge domain (e.g., extracting information from maintenance reports for learning cost-effective maintenance strategies).
- Enabling integrative analysis of both structured and unstructured data in the bridge domain: A large amount of structured and unstructured data are becoming increasingly available in the

bridge domain. However, these bridge data are being used separately. The utilization of the proposed data linking and fusion methods offers opportunities to using structured and unstructured bridge data in integration. The benefit of the integrative use of such data has been manifested in this research in improving the performance of data-driven bridge deterioration prediction. Potential broader benefits are expected, if the methods are applied to integrating all types of data in the domain (e.g., not only integrating NBI data with textual data, but also with health monitoring data, inspection images, etc.). In that case, we could have a unified representation of all the heterogeneous data that covers various aspects of the nation's bridge assets (e.g., bridge condition, serviceability, functionality, etc.) – fully unleashing the power of the data to facilitate bridge asset management.

- Enabling safer, efficient, and cost-effective maintenance of bridges: U.S. bridges only received a grade of C+ (mediocre); 9.1% of the nation's bridges are structurally deficient and 13.6% of them are functionally obsolete (ASCE 2017). It is estimated that the average annual failure rate of the nation's bridges is between 87 and 222, with an expected value of 128 (Cook et al. 2013). In order to eliminate the nation's deficient bridge backlog by 2028, a $20.5 billion annual investment in the construction and maintenance of bridges is needed, while only $12.8 billion is being invested currently (ASCE 2013). The proposed framework, through the extraction, integration, and analysis of both structured and unstructured bridge data from multiple sources, allows decision makers in bridge management to better predict the future deterioration of bridges – offering future opportunities to better understand where to make the best maintenance investments and why those decisions are made, resulting in decisions that are both safe and cost-effective.

- Supporting data analytics applications in the construction and civil infrastructure domain: One direct benefit of the proposed data analytics framework lies in offering new knowledge on how structured and unstructured data can be analyzed in integration for improving the performance of data-driven applications. Such knowledge could be directly transferred to supporting the development of data analytics methods for enhancing deterioration prediction and maintenance decision making for other types of infrastructure (e.g., highway and dam). In that case, the new knowledge offered by this research would benefit the society in better restoring our deteriorating infrastructures. On the other hand, the proposed data analytics framework could be extended to support data analytics for many other applications and purposes in the broader construction and civil infrastructure domain, such as analyzing construction daily reports for supporting predictive project control, analyzing social networking service data for supporting smart community development, etc.

1.7 Publications

This thesis contains material published in the following conference and journal papers:

- Liu, K., and El-Gohary, N. (2019). "A hybrid information fusion method for fusing data extracted from inspection reports for supporting bridge data analytics." *Proc., 2019 Int. Conf. on Computing in Civil Engineering (i3CE)*, ASCE, Reston, VA, 105-112.
- Liu, K., and El-Gohary, N. (2018). "Learning from class-imbalanced bridge and weather data for supporting bridge deterioration prediction." *Proc., 35th Int. Council for Research and Innovation in Building Construction (CIB) W78 2018 Conf.*, Springer, Cham, Switzerland, 749-756.

- Liu, K., and El-Gohary, N. (2018). "Unsupervised named entity normalization for supporting information fusion for big bridge data analytics." *Proc., 25th Int. Workshop on Intelligent Computing in Engineering (EG-ICE)*, Springer, Cham, Switzerland, 130-149.
- Liu, K., and El-Gohary, N. (2018). "Feature discretization and selection methods for supporting bridge deterioration prediction." *Proc., 2018 Construction Research Congress (CRC)*, ASCE, Reston, VA, 413-423.
- Liu, K., and El-Gohary, N. (2017). "Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports." *Automat. Constr.*, 81, 313-327.
- Liu, K., and El-Gohary, N. (2017). "Similarity-based dependency parsing for extracting dependency relations from bridge inspection reports." *Proc. 2017 Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 316-323.
- Liu, K., and El-Gohary, N. (2017). "Ontology-based data integration for supporting big bridge data analytics." *Proc., 2017 Canadian Society for Civil Engineering (CSCE) Annual Conf.*, Vancouver, BC, 1089-1098.
- Liu, K., and El-Gohary, N. (2016). "Semantic modeling of bridge deterioration knowledge for supporting big bridge data analytics." *Proc., 2016 Construction Research Congress (CRC)*, ASCE, Reston, VA, 930-939.
- Liu, K., and El-Gohary, N. (2016). "Ontology-based sequence labelling for automated information extraction for supporting bridge data analytics." *Proc. 2016 Int. Conf. on Sustainable Design, Engineering, and Construction (ICSDEC)*, Elsevier, Amsterdam, Netherlands, 504-510.

CHAPTER 2 - LITERATURE REVIEW

This chapter presents a summary of literature review on semantic modeling and ontology, information extraction, relation extraction, data linking, data fusion, and data-driven bridge deterioration prediction.

2.1 Semantic Modeling and Ontology

2.1.1 Ontology Development Methodologies

Several ontology development methodologies have been well-established within the ontological engineering domain, such as the Toronto Virtual Enterprise (TOVE) Methodology (Fox and Gruninger 1998), SENSUS (Swartout et al. 1996), Methontology (Fernández-López et al. 1997), On-To-Knowledge Methodology (Fensel et al. 2000), and Ontology Development 101 (Noy and McGuinness 2001). Within the civil infrastructure and construction domain, the most notable ontology development methodologies include the methodology by El-Gohary and El-Diraby (2010). The ontology development activities of the above-mentioned methodologies can be generalized into to a process, including: (1) specification, (2) conceptualization, (3) formalization, and (4) implementation (Cristani and Cuel 2005; Sure et al. 2006).

2.1.2 Coverage of Bridge Deterioration Knowledge in Existing Ontologies

Ontologies have been widely developed and applied for knowledge sharing and reuse (Fensel et al. 2000). From an ontology specification perspective, ontologies can be classified into two categories: lightweight and heavyweight (Wong et al. 2012). Lightweight ontologies are presented as glossaries, thesaurus, or taxonomies in which knowledge is represented by a set of controlled vocabularies with or without taxonomy and partonomy structures. Heavyweight ontologies make further efforts towards representing knowledge with richer and more formal relationships and

axioms defined for controlled vocabularies. Following this classification, the review of ontologies that are pertinent to bridge deterioration knowledge is presented in Table 2.1. Kubota and Mikami (2013), Ulieru and Madani (2006), and Halfawy et al. (2005) proposed ontologies that focus on supporting highway bridge design, monitoring, and maintenance. Bień et al. (2007) developed a railway bridge degradation mechanism ontology. El-Diraby and Osman (2011), Osman and El-Diraby (2006), and El-Diraby and Kashif (2005) focused on modeling the design and construction knowledge within the civil infrastructure domain. El-Gohary and El-Diraby (2010) presented a domain ontology for processes in the civil infrastructure and construction domain. BuildingSMART (2014) developed an IFC-Bridge, as an extension to the industry foundation class (IFC), for representing knowledge about bridge elements. Along with the abovementioned heavyweight ontologies, many organizations have also developed and are maintaining several lightweight ontologies, such as the AASHTO Transportation Glossary (AASHTO 2009) and the Transportation Research Thesaurus (TRB 2015).

Table 2.1. Review of relevant ontologies in the construction and civil infrastructure domain.

Sources		Bridge deterioration knowledge concept			
		Bridge element concept	Bridge deficiency concept	Bridge deficiency cause concept	Bridge maintenance action concept
Heavyweight	BuildingSMART (2013)	XX	O	O	O
	Kubota and Mikami (2013)	X	O	O	O
	El-Diraby and Osman (2011)	O	O	O	O
	El-Gohary and El-Diraby (2010)	O	O	O	O
	Bien et al. (2007)	O	X	X	O
	Ulieru and Madani (2006)	X	X	O	O
	Osman and El-Diraby (2006)	O	O	O	O
	El-Diraby and Kashif (2005)	X	O	O	O
	Halfawy et al. (2005)	XX	O	O	O
Lightweight	AASHTO (2009)	X	X	O	O
	NCHRP (2011)	O	O	O	XX
	BTS (2015)	O	O	O	O
	TRB (2015)	X	X	X	O

O = Not cover; X = Rarely cover; XX = Moderately cover.

2.2 Information Extraction

Information extraction (IE) is an automatic process that aims to recognize and extract information of a particular class of entities, relations, or events from natural language text (Hobbs and Riloff 2010; Wimalasuriya and Dou 2010). Existing IE methods can be classified into two primary categories: rule-based methods and ML-based methods (Hobbs and Riloff 2010; Sarawagi 2008; Wimalasuriya and Dou 2010).

2.2.1 Rule-Based Information Extraction

Rule-based IE methods rely on hand-crafted pattern-matching-based rules for guiding the recognition and extraction of target information from unstructured textual data (Nadeau and Sekine 2007; Sarawagi 2008). The pattern-matching-based rules are constructed with syntactic and/or semantic features of text. Outside of the construction domain, many rule-based IE techniques have been proposed (e.g., Appelt et al. 1993; Corro and Gemulla 2013; Elsebai et al. 2009; Fader et al. 2011; Lehnert et al. 1991; Xu et al. 2010). In the construction domain, a limited number of research efforts have focused on developing rule-based IE methods to support various domain-specific tasks. For example, Zhang and El-Gohary (2015) and Zhou and El-Gohary (2015) developed pattern-matching-based rules with both syntactic and semantic features to extract building regulatory information for automated compliance checking. Al Qady and Kandil (2010) developed IE rules with syntactic features to extract concepts from construction contracts.

2.2.2 Machine Learning-Based Information Extraction

ML-based IE methods utilize ML algorithms to automate the rule induction process for IE from text (Nadeau and Sekine 2007; Sarawagi 2008). ML-based IE methods differ from each other

primarily based on the types of ML algorithms used. ML-based IE methods are supervised, semi-supervised, or unsupervised.

2.2.2.1 Supervised Machine Learning-Based Information Extraction

Supervised ML-based IE methods learn from a large set of independent and identically distributed labeled data to recognize and extract information from unlabeled data. A number of supervised ML algorithms have been proposed to support IE, including decision trees (Sekine et al. 1998), support vector machines (Isozaki and Kazawa 2002), structural support vector machines (Tang et al. 2012), hidden Markov models (Bikel et al. 1997), maximum-entropy Markov models (Borthwick et al. 1998), and conditional random fields (CRF) (Lafferty et al. 2001). Among these IE methods, CRF has been widely recognized for supporting IE. This is because: (1) CRF is a graphical model that offers a natural formalism for representing the dependency structures of natural language (Sutton and McCallum 2006); (2) CRF is a discriminative model that captures conditional probabilities to allow for the exploration of a rich set of interdependent features (Sutton and McCallum 2006); and (3) CRF models conditional probabilities globally to prevent the label bias issues (Lafferty et al. 2001).

2.2.2.2 Semi-Supervised Machine Learning-Based Information Extraction

Semi-supervised ML-based IE methods learn from both labeled and unlabeled data to extract information from unlabeled data. Existing semi-supervised ML-based IE methods have been proposed using bootstrapping strategy (e.g., Jiang and Zhai 2007; Liao and Veeramachaneni 2009; Liu et al. 2011; Wu et al. 2009), information-theoretic regularization (e.g., Jiao et al. 2006; Mann and McCallum 2007), or robust representations of unlabeled data as inputs (e.g., Guo et al. 2009; Miller et al. 2004). The bootstrapping strategy relies on an iterative process of adding confidently extracted unlabeled data and re-training a ML model based on the new dataset (Liu et al. 2011).

Thus, it might be prone to noises and requires heuristic determination of stopping criteria (Kuksa and Qi 2010). Information-theoretic regularization aims to regularize learning functions of labeled data through minimizing the entropy of unlabeled data. The regularization process results in a non-concave objective function (Jiao et al. 2006). Concavity is especially important for ML-based IE; otherwise, IE performance could be negatively affected by suboptimal initializations and only reaching to local maxima. Robust representations of unlabeled data are achieved under the cluster assumption, which assumes that if two data points lay in the same cluster, they are likely to have a similar class label (Mann and McCallum 2007). Utilizing the cluster assumption has been proved to be an effective way for developing semi-supervised ML-based IE methods (e.g., Chen and Wang 2011; Mallapragada et al. 2009).

2.2.2.3 Unsupervised Machine Learning-Based Information Extraction

Unsupervised ML-based IE methods learn how each of the unlabeled data should be labeled without learning from labeled data. In the absence of labeled data, some unsupervised ML-based IE methods attempted to group similar entities into a cluster merely based on similarities measured from unlabeled text (e.g., Alfonseca and Manandhar 2002; Etzioni et al. 2005; Nadeau et al. 2006; Shinyama and Sekine 2004). Others also proposed to utilize topic modeling methods, such as probabilistic latent semantic indexing (Hofmann 1999) and latent Dirichlet allocation (Blei et al. 2003), in order to dynamically cluster similar entities (Guo et al. 2009). Because of the existence of statistical dependencies between entities in natural language (Sutton and McCallum 2006), without formally representing and utilizing such dependencies revealed by labeled data, unsupervised ML-based IE methods might be inclined to generate incoherent clusters (Chen 2016).

2.2.3 Semantic Similarity Measures

In the natural language processing (NLP) community, many semantic similarity (SS) measures have been proposed to measure similarities between language units. SS measures can be categorized into: corpus-based and knowledge-based. Corpus-based SS measures, also known as distributional SS measures, quantify the degree of semantic similarity between language units based on their co-occurrences and their linguistic contexts derived from corpus (Harispe et al. 2015; Mihalcea et al. 2006). Existing corpus-based approaches include pointwise mutual information (PMI) (Turney 2001) and latent semantic analysis (LSA) (Landauer et al. 1998), which measure SS between language units in their lexical forms. Corpus-based similarity measuring performance has been improved by also considering the corresponding stems and part-of-speech (POS) tags of information units, in addition to their lexical forms, and by conducting stop-word removal (Harispe et al. 2015; Xie and Liu 2008). Knowledge-based SS measures quantify the degree of semantic similarity between language units according to formal expressions of knowledge, which explicitly define how the information units in comparison must be understood (Harispe et al. 2015; Mihalcea et al. 2006). Knowledge-based SS measures strongly depend on ontologies as knowledge sources (Harispe et al. 2015). Existing knowledge-based approaches include shortest-path approach (Leacock and Chodorow 1998), random-walk approach (Muller et al. 2006), depth-based approach (Wu and Palmer 1994), feature-based approach (Bulskov et al. 2002), and information content approach (Resnik 1995), etc.

2.3 Relation Extraction

2.3.1 Transition-Based Dependency Parsing Model

Dependency parsing (DP) performs a grammatical structure analysis of a sentence to extract dependency relations between “head” words and their corresponding “modifier” words (Buchholz

and Marsi 2006; Chen and Zhang 2015). Existing DP models can be categorized into: graph-based and transition-based (McDonald and Nivre 2007). A graph-based model treats DP as a searching task in which subgraphs are factored, so that the model can search over the space of valid subgraphs to generate the most-likely dependency graph (Chen and Zhang 2015; Nivre and McDonald 2008) (a set of dependency relations for a sentence). A transition-based model treats DP as a classification task, in which a set of configurations generated from an initial configuration are sequentially classified into transition types (indicating word-to-word dependency relations) for extracting dependency relations in a sentence (Chen and Manning 2014; Nivre and McDonald 2008). Transition-based DP models have gained considerable popularity because of their computational efficiency and accurate performance (Chen and Manning 2014; Dyer et al. 2015; Weiss et al. 2015; Choi and McCallum 2013).

The transition-based DP approach was introduced by Nivre (2003). As illustrated in Table 2.2, in the transition-based DP model, a configuration, $\boldsymbol{C} = (\boldsymbol{\sigma}, \boldsymbol{\beta}, \boldsymbol{A})$, is composed of a stack ($\boldsymbol{\sigma}$), a buffer ($\boldsymbol{\beta}$), and a set of dependency arcs/relations ($\boldsymbol{A}$). The stack, $\boldsymbol{\sigma} = [\sigma_i, \ldots \sigma_2, \sigma_1]$, where $i \geq 0$, is a data structure that stores partially-parsed words of an input sentence. The buffer, $\boldsymbol{\beta} = [\beta_1, \beta_2 \ldots, \beta_j]$, where $j \geq 0$, is a data structure that stores the words of the sentence that need to be parsed. The set $\boldsymbol{A}$ is a data structure that stores word pairs that have been parsed with dependency relations. The initial configuration of the input sentence is defined as $\boldsymbol{C} = (\boldsymbol{\sigma} = [Root], \boldsymbol{\beta} = [\beta_1, \beta_2 \ldots, \beta_n], \boldsymbol{A} = \emptyset)$, where $Root$ is a dummy node at the highest level of a dependency graph and $\beta_1, \beta_2 \ldots, \beta_n$ correspond to the words of the sentence (where n is the length of the sentence). The terminal configuration of the sentence is defined as $\boldsymbol{C} = (\boldsymbol{\sigma} = [Root], \boldsymbol{\beta} = \emptyset, \boldsymbol{A})$, where $\boldsymbol{A}$ contains the parsed dependency relations of the sentence. From the initial configuration, the transition-based model predicts a transition type for the current configuration

and generates the next configuration based on the current configuration and the predicted transition type. This process repeats until some terminal configuration has been reached, where the sentence has been completely parsed. Three transition types are defined in the transition-based DP model, including:

- *Shift*: moving β_1 from the buffer $\boldsymbol{\beta}$ to the stack $\boldsymbol{\sigma}$, if $|\boldsymbol{\beta}| \geq 1$.
- *Left-arc*: adding an arc between σ_1 and σ_2, where σ_1 is a head word and σ_2 is a modifier word, and removing σ_2 from the stack $\boldsymbol{\sigma}$, if $|\boldsymbol{\sigma}| \geq 2$.
- *Right-arc*: adding an arc between σ_1 and σ_2, where σ_2 is a head word and σ_1 is a modifier word, and removing σ_1 from the stack $\boldsymbol{\sigma}$, if $|\boldsymbol{\sigma}| \geq 2$.

Table 2.2. Example of a transition-based dependency parsing model.

Transition[a]	Stack	Buffer	Arc (head, modifier)
	[Root]	[The bottom chord connection of truss has severe crevice corrosion]	
S	[Root The]	[bottom chord connection of truss has severe crevice corrosion]	
S	[Root The bottom]	[chord connection of truss has severe crevice corrosion]	
S	[Root The bottom chord]	[connection of truss has severe crevice corrosion]	
S	[Root The bottom chord connection]	[of truss has severe crevice corrosion]	
L	[Root The bottom connection]	[of truss has severe crevice corrosion]	(connection, chord)
L	[Root The connection]	[of truss has severe crevice corrosion]	(connection, bottom)
L	[Root connection]	[of truss has severe crevice corrosion]	(connection, The)
S	[Root connection of]	[truss has severe crevice corrosion]	
S	[Root connection of truss]	[has severe crevice corrosion]	
L	[Root connection truss]	[has severe crevice corrosion]	(truss, of)
R	[Root connection]	[has severe crevice corrosion]	(connection, truss)
S	[Root connection has]	[severe crevice corrosion]	
L	[Root has]	[severe crevice corrosion]	(has, connection)
S	[Root has severe]	[crevice corrosion]	
S	[Root has severe crevice]	[corrosion]	
S	[Root has severe crevice corrosion]	[ ]	
L	[Root has severe corrosion]	[ ]	(corrosion, crevice)
L	[Root has corrosion]	[ ]	(corrosion, severe)
R	[Root has]	[ ]	(has, corrosion)
R	[Root]	[ ]	(Root, has)

[a] S = shift; L = left arch; R = right arch.

2.3.2 Machine Learning-Based Dependency Parsing Methods

Early DP research efforts (e.g., Kurohashi and Nagao 1994; Tapanainen and Järvinen 1997; Oflazer 2003; Elworthy 2000) have focused on developing rule-based DP methods. Rule-based DP methods utilize manually-developed parsing rules to extract dependency relations. More

recently, machine learning-based DP methods have been proposed for automatically classifying configurations into transition types for dependency relation extraction. Some of these efforts have focused on developing probabilistic models (e.g., Eisner 1996; Collins 2003; Samuelsson 2000; Wang and Harper 2004), while others have proposed discriminative approaches with support vector machines (e.g., Kudo and Matsumoto 2003; Yamada and Matsumoto 2003), beam search-based perceptron (e.g., Zhang and Clark 2008; Zhang Nivre 2011), dynamic programming-based perceptron (e.g., Huang and Sagae 2010), or neural networks (e.g., Henderson 2004; Mayberry and Miikkulainen 2005).

In recent years, there has been an increasing number of research efforts focusing on NN-based DP methods (e.g., Chen and Manning 2014; Dyer et al. 2015; Weiss et al. 2015; Alberti et al. 2015; Zhou et al. 2015; Yazdani and Henderson 2015; Cheng et al. 2016; Kiperwasser and Goldberg 2016; Kuncoro et al. 2017; Hashinmoto et al. 2017; Dozat and Manning 2017; Nguyen et al. 2017; Strubell and McCallum 2017; Babbar and Schölkopf 2017). Neural networks have gained popularity in the area of DP for two main reasons. First, as opposed to conventional machine learning-based DP methods (which rely heavily on hand-crafted indicator features), NN-based DP methods can automatically learn the most-useful feature conjunctions and high-order features, which helps avoid feature sparsity and incompleteness issues (Chen and Manning 2014; Pei et al. 2015). Second, DP can benefit from neural networks by learning from NN-based distributed feature representations. Distributed feature representations (also known as word embedding) transform text features [e.g., words and part-of-speech (POS) tags] into real-valued, continuous, and dense vectors, and embed semantically-similar features nearby each other in the vector space (Mikolov et al. 2013). Such representations result in a compact dense feature space, which leads to more efficient, compact, and accurate classifier learning (Chen and Manning 2014). Recent

efforts (e.g., Chen and Manning 2014; Bansal et al. 2014; Guo et al. 2015) have demonstrated that, compared to learning from traditional one-hot feature representations, learning from NN-based distributed feature representations can improve DP performance.

Chen and Manning (2014) is one the first efforts that incorporated neural networks and deep learning into a transition-based DP model (Dozat and Manning 207). They developed a simple, yet relatively accurate and computationally efficient, three-layer feedforward NN architecture for supporting general-domain DP applications. Many NN-based DP methods that used more complex NN architectures have since been developed to further improve the parsing accuracy, such as the recurrent neural network (Kuncoro et al. 2017), the long short-term memory (LSTM) (Kiperwasser and Goldberg 2016), and the bi-LSTM with deep biaffine attention (Dozat and Manning 2017). Compared to the three-layer feedforward NN architecture, these complex architectures were able to marginally improve the parsing accuracy, but at the expense of computational efficiency (see Chen and Manning 2014; Dozat and Manning 2017).

2.3.3 Ensemble Machine Learning Methods

Ensemble machine learning is a learning paradigm that utilizes multiple classifiers to obtain improved performance (reduced variability and increased generalization) that cannot be obtained by any of the constituent classifiers alone (Zhang and Ma 2012; Sun 2013). The most well-established and prominent ensemble learning algorithms include bagging, boosting, stacked generalization, and mixture of experts (Zhang and Ma 2012; Xu et al. 2013). Bagging trains each of the multiple classifiers with a certain percent of instances that are randomly drawn with replacement from the entire training set (Breiman 1996). Boosting sequentially trains a set of classifiers, each of which focuses on learning from the instances that were misclassified by its preceding classifier (Schapire 1990). Adaptive boosting, also referred to as AdaBoost, is a widely

known boosting algorithm. It sequentially trains a set of classifiers, during which the initial classifier is trained with instances sampled based on a uniform distribution and each of the subsequent classifiers is trained with instances sampled according to a weighted distribution, where the weight is updated based on the distribution and training errors of its preceding classifier (Freund and Schapire 1995). Stacked generalization first trains a set of tier-1 classifiers with training instances sampled using cross-validation partitioning, and then trains a tier-2 combiner classifier using the outputs of the tier-1 classifiers as input (Wolper 1992). The combiner classifier aims to learn the misclassification and/or classification patterns to correct the misclassifications generated by the tier-1 classifiers. A mixture of experts trains a set of classifiers (experts) and a gating network that allocates an individual instance to one or several classifiers (Jacobs et al. 1991). The outputs of the selected classifier(s) are then combined through a linear rule to yield a final classification decision for the instance.

2.4 Data Linking

Data linking aims to identify the records – which could be syntactically same, similar, or different – in the same or different data sources that refer to the same entity (i.e., that carries same/similar semantic meaning) (Singla and Domingos 2006; Elmagarmid et al. 2007). Existing data linking methods can be classified into two categories: classification-based and clustering-based.

2.4.1 Machin Learning-Based Data Linking Methods

2.4.1.1 Classification-Based Data Linking Methods

Classification-based methods consider data linking as a binary classification task, which aims to classify the attribute similarity vectors of record pairs into “match” and “non-match” (Christen 2012; Singla and Domingos 2006), where a “match” means that the records should be linked. The

linking methods in this category follow the fundamental principle of the Fellegi-Sunter probabilistic model: record pairs are assumed to be independent and identically distributed, and linking decisions are made independently for each pair (Fellegi and Sunter 1969). The model estimates the m- and u-probabilities (the attribute agreement weights for matches and non-matches) based on training data (i.e., record pairs). For classification, these probabilities are aggregated based on the attribute agreement conditions of the record pair. Besides this probabilistic method, many rule-, distance-, and machine learning (ML)-based linking methods have also been developed. Rule-based methods rely on human-developed classification rules to classify record pairs. For example, in Jiang et al. (2014), a set of rules were developed to link bibliographic data. Distance-based methods compute a distance between a pair of records and compare the distance with a pre-defined threshold value to decide if they are a match or not. For example, in Dey et al. (1998), a weighted distance-based linking method was developed, where attribute weights were solicited from users. ML-based methods learn attribute weights from training examples to capture the linking patterns for classifying record pairs (Christen 2012). A number of supervised ML classification algorithms have been utilized in this regard, including decision trees (Cochinwala et al. 2001; Elfeky et al. 2002), support vector machines (Bilenko and Mooney 2003; Christen 2008), conditional random fields (Gupta and Sarawagi 2009), nearest neighbors (Christen 2008; He et al. 2010), logistic regression (Christen 2008), and random forest (Kejriwal and Miranker 2015).

2.4.1.2 Clustering-Based Data Linking Methods

Clustering-based methods consider data linking as a clustering task, which aims to cluster the records that refer to the same entity into the same cluster (Christen 2012). A number of studies have utilized different clustering algorithms for linking records. For example, a hierarchical clustering algorithm was utilized in (Bilenko et al. 2005) to link online product information.

Correlation clustering algorithms were utilized in many studies (e.g., Soon et al. 2006; Ng and Cardie 2002; Ailon et al. 2008; Elsner and Charniak 2008; Elsner et al. 2000) for supporting various data linking applications. In Hassanzadeh et al. (2009), a number of commonly-used clustering algorithms were implemented, including single-pass clustering algorithms, star clustering, Ricochet family of algorithms, cut clustering, articulation point clustering, Markov clustering, and correlation clustering. Some studies also used clustering algorithms as a post-processing step after classification to deal with transitive closure problems. In these studies, records are represented in graphs, where nodes represent the records and edges represent the links between them. An edge exists between two records, only if they were identified as a match in the classification step. Clustering algorithms, such as CENTER (Haveliwala et al. 2009) and MERGE-CENTER (Hassanzadeh and Miller 2009), are then utilized to partition the graphs into subgraphs to correct the incorrectly-linked records.

2.4.2 Term Similarity Assessment

Term similarity (TS) scoring functions measure to what degree two terms are similar. The commonly-used TS scoring functions are based on either exact comparisons, distances (including edit-, bag-, compression-, syllable alignment-, Jaro-, and Winkler-distances), longest common substrings/sequences, or N-grams. The exact comparison function considers two terms as being similar only if they are exactly the same; otherwise, they are considered completely different. The edit-distance functions, including the Levenshtein (Levenshtein 1966) and Smith-Waterman (Smith and Waterman 1981) edit distances, measure the similarity between two terms based on the minimum number of edit operations (e.g., insertion, deletion, and substitution) needed to convert one term into the other. The bag-distance function measures the similarity between two terms based on the maximum number of distinct letters in them (Bartolini et al. 2002). The

compression-distance function is based on the Kolmogorov complexity theory, under which two terms are similar if one can be significantly compressed given the information of the other (Cilibrasi and Vitányi 2005). The syllable alignment-distance function transforms terms into sequences of syllables based on a set of transformation rules, and measures the similarity between two terms by computing the edit distance between their syllable sequences (Gong and Chan 2006). The longest common substring (LCS) function measures the similarity based on the length of the LCS (the largest number of the same and consecutive letters) in the two terms (Friedman and Sideli 1992). The longest common subsequence (sequence matching) function measures the similarity based on the length of the longest common subsequence (the largest number of the same but not necessarily consecutive letters) in the terms (Bergroth et al. 2000). A variation of the LCS function, the ontology LCS function (which was initially used for ontology alignment), also considers the effect of different substrings on similarity assessment (Stoilos et al. 2005). The N-gram function measures the similarity based on the number of common N-grams (e.g., unigrams, bigrams, and trigrams) in the two terms (Christen 2012; Singla and Domingos 2006; Dey et al. 1998). Variations of the N-gram function include the skip-bigram function (Keskustalo et al. 2003), which considers non-adjacent letters as bigrams, and the positional N-gram function (Keskustalo et al. 2003), which additionally considers the positions of the N-grams. The Jaro-distance function combines the N-gram and edit-distance functions to measure term similarities (Winker and Thibaudeau 1991). The Winkler-distance function (Winkler and Thibaudeau 1991) is similar to the Jaro function, but also considers the effect of common prefixes. For more detailed explanations of these functions, including their equations, the readers are referred to (Christen 2012) and (Elmagarmid et al. 2007).

2.4.3 Concept Similarity Assessment

Concept similarity (CS) scoring functions measure to what degree two concepts are similar. Many semantic similarity (SS) indicators have been developed in this regard, including corpus-based and knowledge-based indicators. Corpus-based SS indicators assess the similarities between concepts based on their cooccurrence rates and their linguistic contexts derived from a text corpus (Harispe et al. 2015; Mihalcea et al. 2006). These indicators require that the concepts should have contextual information (e.g., the preceding and succeeding terms in which the concepts in comparison are embedded). Existing corpus-based indicators include pointwise mutual information (PMI) (Turney 2001) and latent semantic analysis (LSA) (Landauer et al. 1998). Knowledge-based SS indicators assess concept similarities based on the formal expressions of knowledge that explicitly define how the concepts in comparison must be understood (Harispe et al. 2015; Mihalcea et al. 2006). These indicators strongly depend on ontologies as knowledge sources (Harispe et al. 2015). They require the concepts to be mapped to an ontology taxonomy prior to similarity assessment. Existing knowledge-based indicators include the shortest path similarity (Leacock and Chodorow 1998), the random walk similarity (Muller et al. 2006), and the information content-based similarity (Resnik 1995).

2.4.4 Spectral Clustering Methods

Spectral clustering (SC) is a family of the graph partitioning theory-based methods. It aims to find a set of optimal cuts to partition a similarity graph into subgraphs, such that the edges in the same subgraph have higher weights and the edges in different subgraphs have lower weights (Meila 2016; Long et al. 2006; Von Luxburg 2007). In SC, data points $V = \{1, \dots, n\}$ are represented in a similarity graph, $G = (V, E)$, where V_i is a vertex (data point) and E_{ij} is an edge between V_i and V_j. The graph is undirected and weighted, where each edge carries a symmetric, non-negative

similarity S_{ij} (the similarity between the vertices at the two sides of an edge). Then, SC performs eigen-decomposition on a Laplacian matrix L (a square matrix representing the graph, whose elements are derived from an affinity matrix, A, that is same as S), and clusters the data points based on a new matrix that is constructed by the first few leading eigenvectors of L (Lei and Rinaldo 2015). The most-commonly used SC methods include unnormalized (Mohar 1997), normalized (Shi and Malik 2000), and Ng-Jordan-Weiss (NJW) normalized SC (Ng et al. 2002). These methods differ from each other mainly in terms of the graph Laplacian (how to derive L from A): some used an unnormalized graph Laplacian (i.e., $L = D - A$, where D is a degree matrix) or computed generalized eigenvectors from an unnormalized graph Laplacian, while others used a normalized graph Laplacian (i.e., $L = I - D^{-1/2}AD^{-1/2}$, where I is an identity matrix) (Von Luxburg 2007). For a more detailed review of spectral clustering, the readers are referred to Von Luxburg (2007).

2.5 Data Fusion

2.5.1 Named Entity Normalization

Named entity normalization transforms named entities (i.e., concept names) that refer to the same entity into a canonical identifier name (Liu et al. 2012). Existing normalization methods are dictionary-based or machine learning-based, and mainly focus on dealing with the surface-form variations in concept names.

2.5.1.1 Dictionary-Based Named Entity Normalization

Dictionary-based methods rely on established lexicons in domain-specific dictionaries or domain-general knowledge bases (especially Wikipedia) to fuse concept names. The lexicons are used as a look-up source of identifier names. To find an identifier from the lexicons, corpus-based (e.g.,

pointwise mutual information) or knowledge-based (e.g., Jiang-Conrath similarity by Jiang and Conrath 1997) concept similarity assessment methods are used to assess the similarity between a concept name and an identifier. In existing research efforts, domain-specific dictionaries have been utilized for fusing species and organism names (e.g., Pafilis et al. 2013), disease names (e.g., Wei et al. 2016), and biomedical names (e.g., Lee et al. 2016). Wikipedia has been used for supporting named entity normalization-related applications, such as text annotation (e.g., Mihalcea and Csomai 2007), knowledge base construction (e.g., Alhelbawy and Gaizauska 2014), and question answering (e.g., Wang et al. 2017).

2.5.1.2 Machine Learning-Based Named Entity Normalization

Machine learning-based methods use machine learning algorithms to learn how to fuse concept names. A number of supervised algorithms have been used for developing normalization models, including support vector machines (e.g., Magdy et al. 2007), generalized perceptron (e.g., Wagner and Foster 2015), random forests (e.g., Jin 2015), conditional random fields (e.g., Akhtar et al. 2015), feed-forward neural networks (e.g., Leeman et al. 2015), long short-term memory recurrent neural networks (e.g., Han et al. 2019), and Siamese recurrent neural networks (e.g., Fakhraei and Ambite 2018). Some of these models directly predict identifier concept names (e.g., Leeman et al. 2015), and some predict the edit operations (e.g., insert, replace, and delete) needed to convert concept names into their identifiers (e.g., Han et al. 2019). In either case, human-annotated data are required. Because of the challenges in annotating data, several unsupervised normalization methods have been developed (e.g., Yang and Eisenstein 2013; Tahmasebi et al. 2019). Although unsupervised methods do not require annotated data, they need a set of target identifiers as input, in order to compute the similarities between concept names and identifiers (which makes them resemble dictionary-based methods).

2.5.2 Numerical Data Fusion

Numerical data fusion transforms numerical data (e.g., numerical deficiency measures) – either from a single source or different sources and/or at different time points – into a unified representation (Boström et al. 2007). Existing methods mainly use descriptive statistics or fusion theories to conduct data fusion.

2.5.2.1 Descriptive Statistics

Descriptive statistics quantitatively describe the features of a set of data (Mann 1995). The commonly-used descriptive statistics in data fusion include the measures of data central tendency and the measures of data variation. Central tendency measures include arithmetic mean, Bonferroni mean, geometric mean, harmonic mean, Heronian mean, power mean, median, and mode. Variation measures include coefficient of variation, mean absolution deviation, range, standard deviation, and variance. For a detailed description of these measures, the readers are referred to Mendenhall and Sincich (2016). Although descriptive statistics are simple, they have been used in some data fusion applications and achieved certain levels of success. For example, using a set of descriptive statistics, Wimmer et al. (2008) fused audio and video features for emotion recognition; Zhang (2015) fused water-depth data and bathymetry data for creating benthic habitat maps; and Varga et al. (2018) fused pixel-level normalized difference vegetation indexes across time for land cover analysis.

2.5.2.2 Data Fusion Theory

Several data fusion theories have been developed, including the Dempster-Shafer theory (Shafer 1976), fuzzy set theory (Zadeh 1965), possibility theory (Zadeh 1978), and rough set theory (Pawlak 1992). The Dempster-Shafer theory assigns a belief mass to a fused value (which could

be a single number, interval, or set) based on the strength of the evidence supporting this value. In the presence of evidence from multiple sources, it uses a joint belief mass function to fuse the belief masses, where the function considers both the agreement and conflict levels of the evidence. It selects the fused value that has the largest belief mass to represent data from multiple sources. The fuzzy set theory is a theoretical reasoning scheme, which uses the partial set memberships of data to allow for imprecise, rather than crisp, reasoning (Khaleghi et al. 2013). The memberships of imprecise data to a fused value are quantified using a membership function (e.g., piecewise linear functions and Gaussian distribution function), and are then fused using an aggregation function (e.g., averaging, conjunctive, and disjunctive functions). The fused value that has the highest aggregated membership degree is used to represent imprecise data from multiple sources. The possibility theory, as an extension of the fuzzy set theory, was developed to further deal with incomplete data using possibility and necessity measures, which quantify the plausibility and the certainty of a fused value given incomplete data, respectively (Destercke et al. 2009). The rough set theory could be applied for data fusion by using lower and upper approximations to find a fused value that has the highest approximation accuracy for representing data from multiple sources. Despite being theoretically-applicable, this theory has been rarely used in data fusion (Khaleghi et al. 2013).

2.6 Data-Driven Bridge Deterioration Prediction

2.6.1 Data-Driven Bridge Deterioration Prediction Methods

Existing data-driven bridge deterioration prediction methods/models can be classified into three categories: deterministic, stochastic, and artificial intelligence (AI)-based (Morcous et al. 2002). Deterministic methods/models use a mathematical formulation to capture the relationship between the conditions of bridges and the factors that affect the deterioration of bridges for predicting the

future bridge conditions (Morcous et al. 2002). Most of the existing deterministic models were developed using regression techniques. For example, to predict the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures), Hatami and Morcous (2011) developed a nonlinear regression model using the NBI data from the Nebraska Department of Transportation (DOT); Chang et al. (2017) developed a logistic regression model with the least absolute shrinkage and selection operator (LASSO) using the NBI data from the Wyoming DOT; Goyal et al. (2017) developed a proportional hazards regression model using the NBI data from the North Carolina DOT; and Lu et al. (2019) developed an ordinal logistic regression model using the NBI data from the North Dakota DOT.

Stochastic methods/models use one or more random variables to capture the uncertainty and randomness of the deterioration process of bridges for predicting the future bridge conditions (Morcous et al. 2002). The majority of the existing stochastic models were developed using the Markov-chain process. For example, Morcous (2006) developed a first-order Markov-chain model using the deck condition rating data from the Ministère des Transports du Québec (MTQ) to predict the future ratings of decks. Wellalage et al. (2014) developed a Metropolis-Hasting algorithm-based Markov-chain model using the timber deck condition rating data from the state of Victoria of Australia to predict the future ratings of timber decks. Fang and Sun (2018) developed a Weibull distribution-based semi-Markov model using the bridge inventory data from the City of Shanghai to predict the condition ratings of bridges. Abdelkader et al. (2019) developed a semi-Markov model using the concrete deck condition rating data from the Quebec Province of Canada to predict the ratings of concrete decks.

AI-based methods/models use computational intelligence (particularly machine learning) to learn from bridge data to predict the future conditions of bridges. For example, Huang (2010) used

backpropagation-based multilayer perceptron neural networks to learn from the bridge inventory data from the Wisconsin DOT to predict the condition ratings of decks. Creary and Fang (2015) used artificial neural networks (ANNs) to learn from the NBI data from the Connecticut DOT to predict the condition ratings of the primary bridge components. Contreras-Nieto et al. (2016) used ANNs to learn from the NBI data from the Oklahoma DOT to predict the condition ratings of the superstructures of steel and prestressed concrete bridges. Lim and Chi (2019) used the extreme gradient boosting algorithm to learn from the bridge inventory data provided by the Korean Bridge Management System to predict the condition ratings of deck damages.

2.6.2 Recurrent Neural Network

A recurrent neural network (RNN) is a type of artificial neural network, which extends the standard feed-forward neural network to allow for the modeling of sequential data. For a timestep in a sequence, an RNN updates its hidden state at the timestep based on the current input and the previous hidden states, and makes a prediction for the input based on the updated state (Sutskever et al. 2011). A standard RNN is mathematically formalized as follows (Graves et al. 2013): for x_t in an input sequence $\mathbf{x} = (x_1, x_2, \cdots x_t)$, the network computes its corresponding hidden state as $h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$ and its output as $y_t = W_{hy}h_t + b_y$, where W, b, and $\mathcal{H}$ denote weight matrices, a bias vector, and a hidden layer activation function, respectively. The current hidden state depends on its previous states such that these hidden states collectively serve as the memory of the network, which allows the network to capture the temporal dynamics that connect data over time for performing sequential prediction (Che et al. 2017). The standard RNN, in practice, is very insufficient in learning long-range dependencies with gradient descent, because the learning error vanishes as it gets propagated back to the network (Hochreiter 1998; Trinh et al. 2018). Two prominent variations of the standard RNN, which incorporate gating mechanisms,

have been developed to address the vanishing gradient problem: long short-term memory RNN (Hochreiter and Schmidhuber 1997) and gated recurrent unit RNN (Cho et al. 2014). Despite being able to capture long-range dependencies, these variants are more computationally-intensive compared to the standard RNN (Chiu and Nichols 2016; Li et al. 2019). Therefore, when dealing with short sequences that have short-range dependencies, the standard RNN is effective enough in terms of both computational efficiency and performance (Tang et al. 2019).

2.6.3 Manifold Learning

It is challenging for machine learning to analyze high-dimensional data efficiently. Such data are assumed to lie on or near a low-dimensional manifold (i.e., subspace) embedded within a high-dimensional space (Zhu et al. 2018). Manifold learning, also known as dimensionality reduction, aims to discover such a manifold for embedding high-dimensional data, which are usually sparse at the same time, into a low-dimensional dense space (Costa and Hero 2004). Prominent manifold learning methods include principal component analysis (Jain and Dubes 1998), multidimensional scaling (Cox and Cox 2000), isometric feature mapping (Tenenbaum et al. 2000), locally linear embedding (Roweis and Saul 2000), and Laplacian eigenmaps (Belkin and Niyogi 2002). Principal component analysis conducts orthogonal transformation to project high-dimensional data into a low-dimensional orthonormal space that maximizes the variance of the data. Multidimensional scaling aims to find a low-dimensional space, such that the Euclidean distance matrix of the embedded data in this space is similar to the matrix of the original data in the high-dimensional space. These two methods assume that the manifold is linear. Isometric feature mapping, which is a nonlinear generalization of the multidimensional scaling method, uses the geodesic distance (instead of the Euclidean distance) to capture the nonlinearity of the manifold. Locally linear embedding assumes that a nonlinear manifold is locally linear and, thus, represents each data

instance in a high-dimensional space using the weighted linear combination of its neighbors. It embeds the data into a low-dimensional nonlinear manifold that preserves the weights (Sual et al. 2006). Laplacian eigenmaps is closely related to locally linear embedding as they share the same objective function (Ghodsi 2006), but differs from it mainly in terms of how to compute the weights. Laplacian eigenmaps computes the weights based on the distance between two data instances (i.e., $W_{ij} = e^{-\|x_i - x_j\|^2/s}$, where x_i and x_j are data instances in a high-dimensional space, W_{ij} is the weight between them, and s is a free parameter), whereas locally linear embedding computes the weights based on how well they can reconstruct a data instance from its neighbors (Ghodsi 2006).

2.6.4 Data Imbalance

A dataset is imbalanced if the number of data instances in one class is greater than that in another class (Longadge and Dongre 2013). Data imbalance negatively affects the performance of standard machine learning algorithms. These algorithms usually assume that the class distributions of data are balanced and/or the costs of misclassifications are equal (He and Garcia 2008). Therefore, when learning from imbalanced data, they cannot properly capture the distribution characteristics of the data and would lead to "imbalanced" performance – the performance of the majority classes is high, and the performance of the minority classes is very low (Ganganwar 2012). Two main approaches have been developed to address the negative impacts caused by data imbalance: data sampling and cost-sensitive learning. To balance a dataset, data sampling over-samples the dataset by increasing the number of instances in the minority classes or under-samples the dataset by decreasing the number of instances in the majority classes. The commonly-used over-sampling techniques include random over-sampling, synthetic minority over-sampling technique (SMOTE) (Chawla 2002), borderline-SMOTE (Han et al. 2005), and adaptive synthetic sampling (He et al.

2008). Over-sampling replicates data instances and, hence, it would largely decrease the computational efficiency of a learning algorithm and would make the algorithm overfitted (Ganganwar 2012). The commonly-used under-sampling techniques include random under-sampling, one-sided selection (Kubat and Matwin 1997), neighborhood cleaning rule (Jorma 2001), and condensed nearest neighbor rule (Batista et al. 2004). Under-sampling discards data instances that could be important in the model learning process, which would undermine the overall performance of a machine learning algorithm (Ganganwar 2012). Cost-sensitive learning deals with data imbalance by taking the costs of misclassifications into consideration. The underlying principle of cost-sensitive learning is that: the cost of incorrectly classifying instances in the minority classes is much higher than the cost of incorrectly classifying instances in the majority classes (He and Garcia 2008). The most common practice of implementing cost-sensitive learning is to introduce a weighting factor to the loss function of a machine learning classifier (Lin et al. 2017), so that the factor can regulate the costs of misclassifications in the aforementioned way to make a learning algorithm focus on learning from instances in the minority class to improve the performance.

CHAPTER 3 – SEMANTIC DATA MODELING AND ONTOLOGY DEVELOPMENT

This chapter presents the proposed bridge deterioration knowledge ontology (BridgeOnto). The BridgeOnto development, coding, and evaluation (Research Task #2) are presented in this chapter. The application-oriented validation of the ontology – evaluating the ontology in supporting information and relation extraction from textual bridge inspection reports – was conducted as a part of Research Task #3 in Chapter 4 and Research Task #4 in Chapter 5; and is presented in these chapters.

3.1 BridgeOnto Development and Coding

3.1.1 BridgeOnto Development and Evaluation Methodology

The ontology development methodology by El-Gohary and El-Diraby (2010) was benchmarked. The BridgeOnto development methodology is summarized as follows:

- *Domain, purpose, intended users, and scope definition*: These fundamental scope descriptions were defined (as per Table 3.1) and utilized as guidance throughout the BridgeOnto development process.
- *Competency questions (CQs) development*: A competency question (CQ) is expressed in the form of a natural language sentence that shows a pattern for a type of questions that an ontology must be able to answer (Fox and Gruninger 1998). CQs serve as functional requirements to ontologies. A set CQs for formulating the functional requirements to the BridgeOnto were developed and are discussed in Section 3.2.1.
- *Concept hierarchy construction*: A concept hierarchy was constructed using two main iterative steps: (1) extracting key concepts from identified concept sources, and (2) organizing the extracted concepts into a concept hierarchy. The key concepts were identified, extracted, and

defined based on a comprehensive review of concept sources. Table 3.2 shows a partial list of the concept sources and their concept coverage distributions. A combination of top-down and bottom-up hierarchy construction approaches were used to avoid the inclusion of unnecessarily too detailed specific concepts and/or less-meaningful high-level concepts. The top-down approach first defines the most general concepts and then specifies their subconcepts; whereas, the bottom-up approach begins with defining the most specific concepts and then groups them into high-level concepts (Noy and McGuinness 2001).

- *Multimodality modeling*: The concept hierarchy was reclassified based on different modality views for representing the polymorphic and multifaceted nature of bridge deterioration knowledge. Different modality views are shown Section 3.1.2.
- *Relation modeling*: Three major types of relations were captured: (1) "is-a" relationship to characterize sub-superordinate relationships, (2) "is-part-of" relationship to decompose concepts into their constituent parts, and (3) cross-concept relationship to establish non-hierarchical relationships with semantic meanings between concepts.
- *Ontology capturing*: Ontology capturing involves conceptualization. The BridgeOnto conceptualization identified formal domain terms for representing the concepts and relations constructed and modeled in the previous steps. The conceptualization of the BridgeOnto is presented in Section 3.1.2.
- *Ontology coding*: The ontology was coded in Web Ontology Language (OWL) using Protégé 3.4.5 (Protégé 2016). The coding details are discussed in Section 3.1.3.
- *Ontological model evaluation*: The BridgeOnto was verified by answering CQs and conducting automated consistency and redundancy checking. It was validated using human expert interviews and application-based validation. The verification and expert interview validation

processes and results are presented in Section 3.2. The application-based validation – evaluating the ontology in supporting information and relation extraction from textual bridge inspection reports was conducted as a part of Research Task #3 in Chapter 4 and Research Task #4 in Chapter 5.

Table 3.1. Domain, purpose, intended users, and scope definition.

BridgeOnto attribute	Definition
Domain	• Bridge deterioration knowledge generated during bridge inspection and maintenance processes
Purpose	• Presenting a domain-specific, unambiguous, and formalized representation of bridge deterioration knowledge • Providing a semantic model that facilitates the recognition, extraction, and representation of key data and information (i.e., the data and information defined by bridge deterioration knowledge) from unstructured textual bridge inspection reports
Intended users	• Bridge domain information users (e.g., bridge engineers, bridge owners, bridge project managers, bridge inspectors, bridge maintenance workers, software developers for bridge management systems, consultants, and regulators)
Scope	• Covering knowledge about bridge elements • Covering knowledge about bridge deficiencies • Covering knowledge about bridge deficiency causes • Covering knowledge about bridge maintenance actions • Covering knowledge about bridge deterioration mechanism as reflected by the above-mentioned knowledge context dimensions

Table 3.2. Concepts and corresponding concept sources.

Source	BridgeOnto main concepts													
	Bridge Element	Bridge Deficiency [a]					Bridge Deficiency Cause [b]					Bridge Maintenance Action [c]		
		D1	D2	D3	D4	D5	C1	C2	C3	C4	C5	M1	M2	M3
FHWA 1995	x	o	o	o	o	o	o	o	o	o	o	o	o	o
FHWA 2012	xx	xx	xx	xx	xx	xx	x	x	x	x	x	o	o	o
AAHSTO 2002	x	x	x	x	x	o	o	o	o	o	o	o	o	o
AASHTO 2010	xx	xx	xx	xx	xx	xx	o	o	o	o	o	o	o	o
State DOTs*	xxx	xxx	xxx	xxx	xxx	xxx	xx	xx	xx	xx	xx	o	o	o
USACE 1995	o	xxx	o	o	o	o	xx	o	o	o	o	xx	o	o
Bien et al. 2007	o	x	x	x	x	x	x	x	x	x	x	o	o	o
Woodson 2009	o	xx	o	o	o	o	xx	o	o	o	o	xx	o	o
Delatte 2009	o	x	o	o	o	o	xx	o	o	o	o	xx	o	o
Hobbs 2011	o	x	o	o	o	o	xx	o	o	o	o	x	o	o
PCA 2002	o	x	o	o	o	o	xx	o	o	o	o	o	o	o
ACI 2002	o	o	o	o	o	o	o	o	o	o	o	xx	o	o
ASCE 2001	o	xx	xx	xx	xx	xx	o	o	o	o	o	o	o	o
Bijen 2003	o	x	x	x	x	o	x	x	x	x	o	x	x	x
Gimmer 1984	o	o	o	o	xx	o	o	o	o	x	o	o	o	o
AASHTO 2007	x	x	x	x	x	x	x	x	x	x	x	xx	xx	xx
State DOTs**	x	x	x	x	x	x	x	x	x	x	x	xxx	xxx	xxx

O = Not cover; X = Rarely cover; XX = Moderately cover; XXX = Cover;
a: The index follows the index in the Figure 3.3;
b: The index follows the index in the Figure 3.4;
c: The index follows the index in the Figure 3.5;
*: The bridge inspection manuals from state DOTs, including Alabama DOT, California DOT, Delaware DOT, District of Columbia DOT, Hawaii DOT, Illinois DOT, Indiana DOT, Louisiana DOT, Minnesota DOT, New York DOT, Ohio DOT, Oregon DOT, Virginia DOT, and Washington DOT;
**: The bridge maintenance manuals from state DOTs, including Alabama DOT, Alaska DOT, California DOT, Delaware DOT, Georgia DOT, Indiana DOT, Iowa DOT, Kentucky DOT, Michigan DOT, Ohio DOT, Utah DOT, and Washington DOT.

3.1.2 Proposed Bridge Deterioration Knowledge Ontology – BridgeOnto

The BridgeOnto aims to provide a domain-specific, unambiguous, and formalized representation of bridge deterioration knowledge. It consists of concepts and inter-concept relations. Concepts represent the "things" that describe bridge deterioration knowledge. Relations define the inter-links between concepts to reflect how they interact.

3.1.2.1 Main BridgeOnto Model

The main BridgeOnto model, which represents the most abstract concepts of the BridgeOnto, is shown in Figure 3.1. At the highest level of abstraction, a "bridge" is composed of a "bridge element", is affected by a "bridge deficiency" that is caused by a "bridge deficiency cause" and that is maintained by a "bridge maintenance action", and has a "bridge attribute". A "bridge deficiency cause" shows existence at the "bridge". A "bridge attribute" could be a "bridge element attribute", a "bridge process attribute", or a "bridge deficiency attribute". A "bridge element attribute" characterizes a "bridge element"; a "bridge process attribute" defines the context dimensions (e.g., inspection/maintenance date and methods) in which a "bridge deficiency" and/or a "bridge maintenance action" exists; and a "bridge deficiency attribute" defines the attributes, including quantity, severity, and onset date, etc., of a "bridge deficiency".

Following the Nation Bridge Inspection Standards (NBIS), a "bridge" is "a structure including supports erected over a depression or an obstruction, such as water, highway, or railway, and having a track or passageway for carrying traffic or other moving loads, and having an opening measured along the center of the roadway of more than 20 feet between undercopings of abutments or spring lines of arches, or extreme ends of openings for multiple boxes; it may also include multiple pipes, where the clear distance between openings is less than half of the smaller

contiguous opening ” (NBIS 2004). A “bridge element” is a bridge primary structural component or a bridge secondary component, which includes a set of elements that are made of the same type of material to form identifiable and performance-similar portions of a bridge and is commonly encountered in bridge inspection and maintenance to determine the overall condition and safety of a bridge (AASHTO 2010; FHWA 2012; NCHRP 2014). Bridge primary structural components are bridge elements designed to resist flexure and distribute both primary live loads and dead loads. Bridge secondary components are bridge elements that do not carry calculated live loads. A “bridge deficiency” is a defect that makes a bridge element to fail meet quality requirements and therefore makes a bridge less capable or less desirable to use (AASHTO 2009; FHWA 2012). A “bridge deficiency cause” is the presence of a certain reaction or phenomena that leads to a deficiency on a bridge structure (e.g., an Alkali-Silica reaction that causes cracking or scaling in concrete bridge elements). A “bridge maintenance action” is an act performed to care for and maintain a bridge and its associated features as nearly as possible to their current, as-constructed, or subsequently improved conditions, so it substantially retains its original use and function (AASHTO 2007; AASHTO 2011; Rogers 2006; WiSDOT 2015). The subconcept hierarchies of the most abstract concepts of the BridgeOnto are presented and discussed in more details in the following subsections.

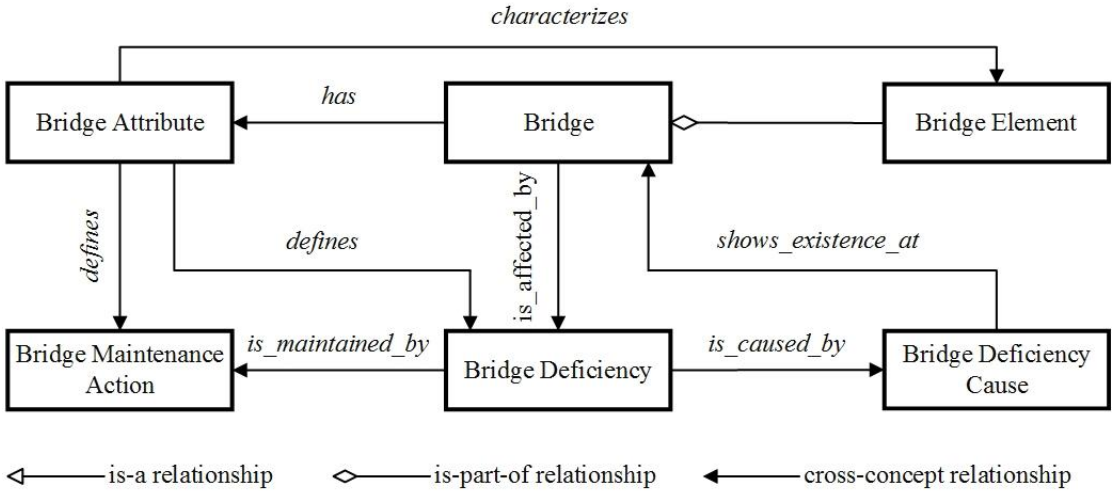


Figure 3.1. Main BridgeOnto model.

3.1.2.2 Bridge Element Hierarchy

The knowledge about bridge elements is an important component of bridge deterioration knowledge, because (1) recently-developed bridge inspection practices have emphasized on reliability-based inspection strategies at the element level (Washer et al. 2014), and (2) recent efforts for developing a uniform bridge maintenance actions database system have stressed on reporting and interpreting bridge maintenance actions at the element level (NCHRP 2011). Driven by such needs, the commonly-used bridge elements have evolved from the Nation Bridge Inventory System (NBIS) elements, to the AASHTO Commonly Recognized (CoRe) Structural Elements, and to the most recent National Bridge Elements (NBE), Bridge Management Elements (BME), and Agency Developed Elements. To support the extraction of information from textual reports that are guided by the abovementioned bridge element systems, the BridgeOnto should capture the key bridge element concepts defined by these systems integrally, in order to ensure sufficient coverage, classification, and applicability of the bridge element hierarchy. Figure 3.2 shows a partial view of the bridge element hierarchy. A partial list of the concepts with their corresponding sources is shown in Table 3.2.

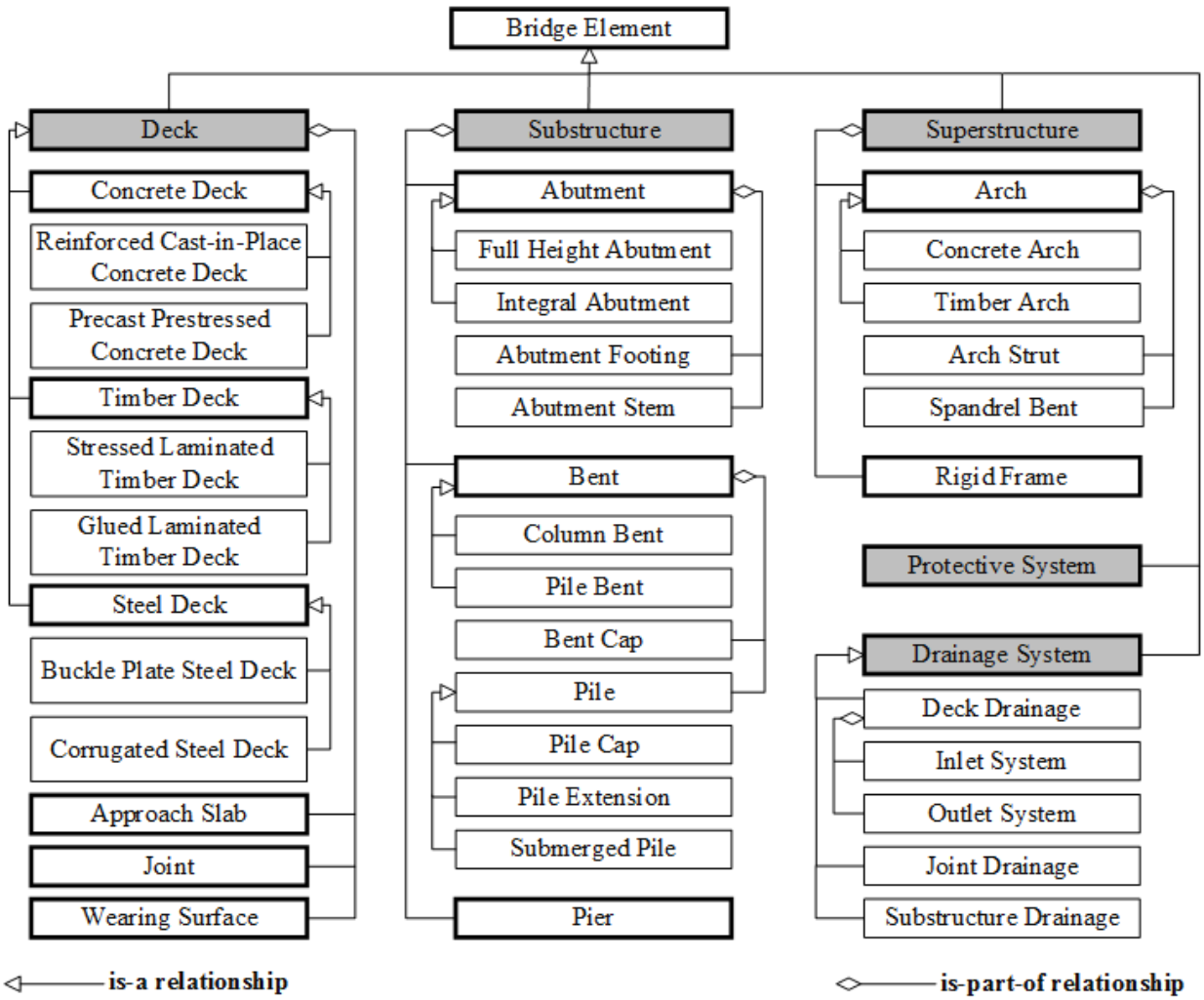


Figure 3.2. Bridge element hierarchy (partial).

In the BridgeOnto, the bridge element hierarchy (including both taxonomy and partonomy) defines the types and parts of bridge elements. At the highest level of abstraction, a "bridge element" could be a "deck", a "substructure", a "superstructure", a "drainage system", or a "protective system". A "deck" is a bridge subsystem that transfers loads to other bridge components (i.e., bridge superstructure) and provides a smooth and safe riding surface to traffics. A "superstructure" is a bridge subsystem that transmits loads directly to bridge substructures. A "substructure" is a bridge subsystem that supports a bridge "superstructure" by transmitting loads into ground. A "drainage system" is a bridge subsystem that removes water from bridge structures. A "protective system"

is also a bridge subsystem that precludes undesirable matters (e.g., moisture and deicing chemicals) from entering bridge structures. Capturing both the width and the depth of bridge element concepts is important, because it allows for representing bridge deterioration knowledge in different perspectives. For example, deterioration mechanisms of different parts of bridge elements (e.g., deck and wearing surface) and of different types of a bridge element (e.g., concrete wearing surface and timber wearing surface) present more views of deterioration knowledge that cannot be presented by a bridge element hierarchy that only captures either width or depth.

3.1.2.3 Bridge Deficiency Hierarchy

Figure 3.3 shows a partial view of the bridge deficiency hierarchy. A partial list of the concepts with their corresponding sources is shown in Table 3.2. In the BridgeOnto, the upper-level classification of the bridge deficiency hierarchy is based on the most common bridge materials. This is because material is a major factor that affects the structural performance of bridges and decides the types of deficiencies they can be affected with (Farhey 2014). According to the (FHWA 2015), approximately 66%, 30%, and 3% of the nation's bridges are constructed with concrete, steel, and timber, respectively. Masonry, although is rarely used in modern bridge constructions, is used in many old stone bridges that are still in service and require inspections and maintenances (FHWA 2012). Fiber reinforced polymer is gaining popularity in bridge construction and maintenance (Cerullo 2013). Based on this classification criterion, a "bridge deficiency" could be a "concrete bridge element deficiency", "fiber reinforced polymer bridge element deficiency", a "masonry bridge element deficiency", a "steel bridge element deficiency", or a "timber bridge element deficiency".

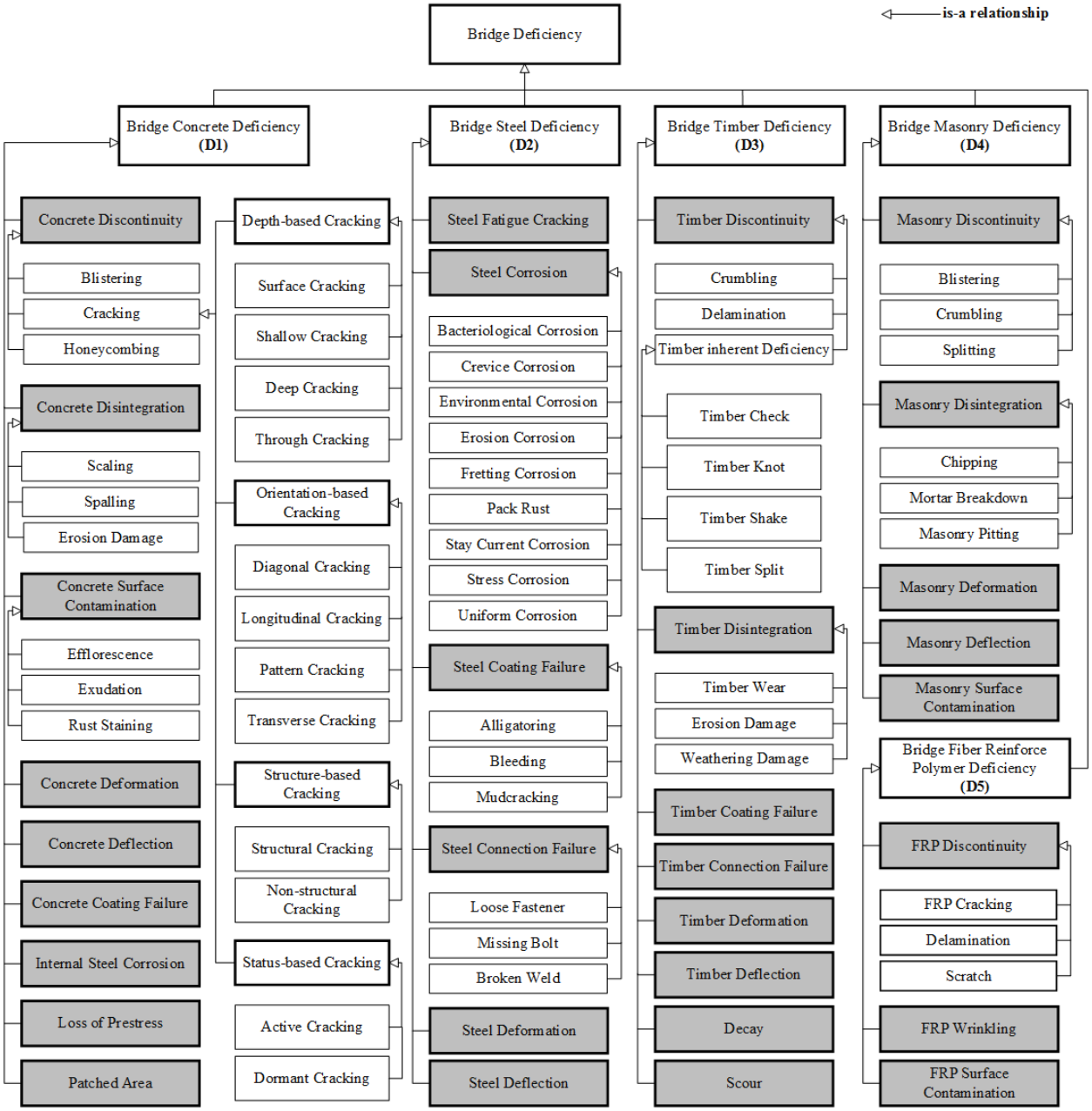


Figure 3.3. Bridge deficiency hierarchy (partial).

Seven primary types of subconcepts under this highest-level classification are defined based on the visual appearances of deficiencies. Visual inspection (VI) is a predominant nondestructive evaluation technique for bridge inspection (FHWA 2001), during which bridge inspectors are responsible to identify bridge deficiencies visually (FHWA 2012). Textual bridge reports document and reflect inspected bridge deficiency findings on a visual-appearance base. Therefore,

the BridgeOnto should capture bridge deficiency concepts in a way that is consistent with how they were documented, in order to be sufficient in applicability. As such, the seven primary types of subconcepts are presented as follows:

- *Discontinuity*: Discontinuity is defined as the non-designed separation in the continuity of structural materials. For example, under the "bridge concrete deficiency" concept, a "concrete discontinuity" could be a "cracking", a "delamination", or a "honeycombing", etc. Similarly, under the "bridge timber deficiency" concept, a "timber discontinuity" could be a "timber check", "timber shake", or a "timber split", etc. The subconcepts of discontinuity are same in terms of visual appearance, because they all manifest as separations on bridge elements with visible linear fractures or fissures in various directions.
- *Disintegration*: Disintegration is defined as the gradual and continuing removal/loss of small particles or fragments from bridge structure surfaces. A "concrete disintegration" could be manifested as a "chalking", a "scaling", a "spalling", or an "erosion damage", etc. The visual appearances of such deficiencies are powdered structure surfaces, polished structure surfaces, or holes on structure surfaces that expose the structure inner layers (e.g., reinforcement in concrete).
- *Surface contamination*: Surface contamination is defined as the precipitation of undesired dirty and/or colored substances on structure surfaces through pores or openings. Surface contamination is commonly found on concrete and masonry bridge elements. For example, a "concrete surface contamination" could be an "efflorescence", a "rust staining", or an "incrustation", etc. Surface contamination deficiencies often indicate more complex deteriorations of bridge elements. For example, the "efflorescence" – white surface deposits – may suggest the chloride contamination of concrete elements. The "rust staining" – red or

orange surface coats – may indicate the corrosion of the internal reinforcement of reinforced concrete elements.

- *Coating failure*: Coating failure, as the name indicates, is the failure of a bridge element protective system. The coating of bridge elements is visually easy to be identified; hence, the coating failures are visually-distinctive compared to other deficiencies. Some coating failures, such as "wrinkling" (i.e., unsmooth and crinkled paint surface), result from construction errors (e.g., excessive paint). Some coating failures, such as "saponification" (i.e., soft residue caused by chemical reactions between concrete surface and paint), are induced by improper selection of coating materials (e.g., oil-based paint for coating concrete). And, other coating failures, such as "coating erosion" (i.e., gradual removal of coating substances from element surfaces), are resulted from physical abrasions from the service environment (e.g., rain, hail or traffics abrasions).
- *Deformation*: Deformation is defined as the elastic or inelastic distortion of a bridge element. The deformation is visually identified as an out-of-shape configuration in a bridge element, compared to as-designed or as-constructed configurations. For example, a "steel deformation" could be a "bending", a "bulking", a "twisting", or a "faulting" due to repeated bending, compressive, torsional, or shear forces; or could be a "rutting" or a "surface depression" resulting from traffic loads.
- *Deflection*: Deflection is defined as the movement of bridge elements under loads. Deflection deficiencies manifest themselves as displacement or misalignment of bridge elements. For example, a "timber deflection" could be a "vertical timber deflection", a "lateral timber deflection", or a "rotation timber deflection" according to the direction to which a bridge element moves.

- *Corrosion*: Corrosion is defined as the destruction of metals from a refined form to a more stable form through oxidation. A rusting metal surface is the most prominent visual indicator of corrosion. In the BridgeOnto, a "corrosion" could be a "bacteriological corrosion" due to the existence of various organisms, a "stay current corrosion" due to electricity from surrounding structures, or a "pack rust" due to mating metal surfaces, etc.

3.1.2.4 Bridge Deficiency Cause Hierarchy

In the BridgeOnto, multimodality views for the bridge deficiency cause hierarchy are presented. Different modality views for the bridge deficiency cause hierarchy intend to capture its importance in bridge deterioration knowledge modeling, because (1) proper bridge inspection practices require the understanding, identification, and recording of the causes of bridge deficiencies (FHWA 2012); and (2) permanent and reliable bridge maintenance actions require the treatment of the root-causes of the deficiencies (AASHTO 2007). Figure 3.4 presents a partial view of the main modality view, and Figure 3.5 shows partial views of the secondary modalities. A partial list of the concepts with their corresponding sources is shown in Table 3.2.

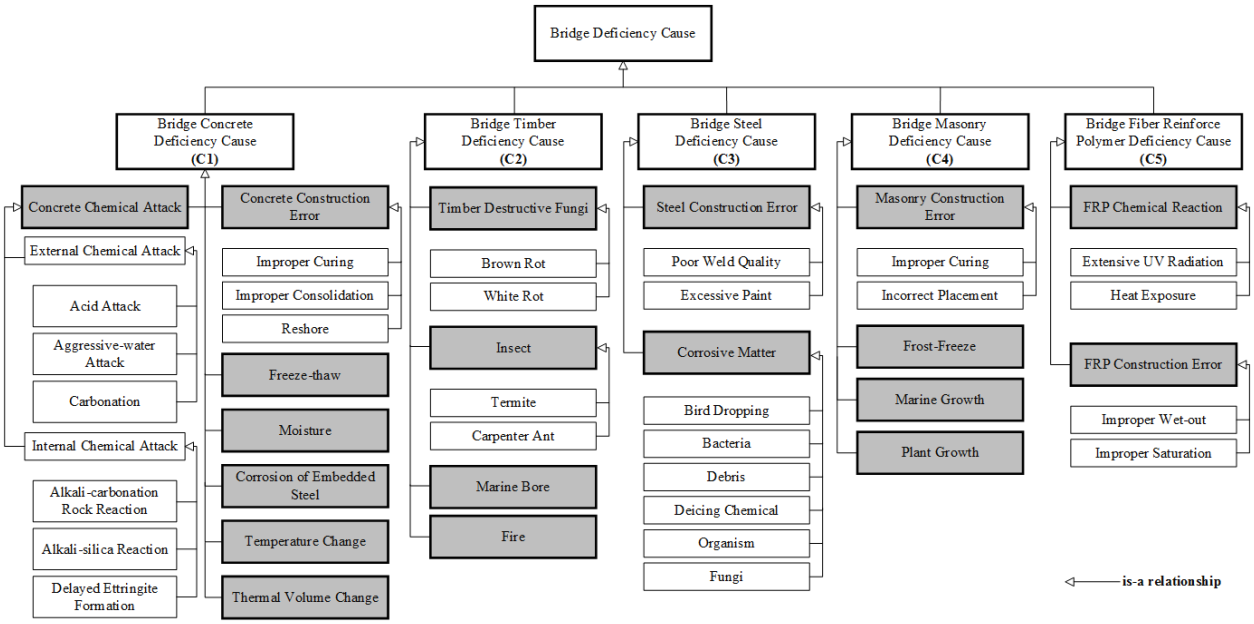


Figure 3.4. Bridge deficiency cause hierarchy (partial).

3.1.2.4.1 *Main Modality View*

In the main modality view, bridge deficiency cause concepts are classified based on primary bridge materials, including concrete, steel, timber, masonry, and fiber reinforced polymer, because different types of material exhibit different properties that make them show varying levels of resistance to certain deficiency causes. As a result, some deficiency causes are considered as critical inducing factors to a particular type of bridge element material, while not to others. For example, concrete, which often contains highly alkaline cement, can quickly deteriorate as the PH of chemical agents it is exposed to decreases from 6.5 (PCA 2016). Also, hydraulic cement concrete can hardly resist a chemical agent with a PH of 3 or lower (PCA 2016). However, plywood, as a type of timber material, has better resistance to chemical agents. It was shown that plywood can barely be affected when exposed to a chemical reagent whose PH is between 2 and 10 (APA 2016). In such case, acid attack is a critical concrete bridge element deficiency cause, but

not (in most cases) a timber bridge element deficiency cause. To this end, in the main modality view, five main subconcepts of "bridge deficiency cause" are defined:

- *Concrete bridge element deficiency causes*: The majority of bridge concrete deficiencies are caused by "accidental loading", "concrete chemical attack", "concrete construction error", and other related causes from concrete element service environments such as "moisture", "freeze-thaw", and "temperature change". An "accidental loading" characterizes massive forces that are applied to concrete elements for a short duration, such as forces due to earthquake and hurricane impacts and vehicle collisions to the concrete bridge elements. A "concrete chemical attack" is one the most predominate causes of concrete bridge elements, because concrete is a highly alkaline material that shows lower resistance to acid attacks, alkali-silica reactions, and aggressive-water (i.e., water with dissolved mineral concentrations) attacks, etc. A "concrete construction error", although may not directly contribute to the failure of bridge elements, could negatively affect concrete properties (e.g., strength and elasticity) and thus open chances for concrete deficiencies.
- *Steel bridge element deficiency causes*: The main types of bridge steel deficiency causes are "accidental loading", "corrosive matter", and "steel construction error". The accidental loading share exactly same characteristics as that causing deficiencies in other types of bridge material. The most recognizable deficiency of steel – corrosion – is caused by the exposure of unprotected or even protected steel material to corrosive matters. "Corrosive matters" are those substances or living things that contribute to or expedite the corrosion of steel. For example, excessive moisture, oxygen, and/or deicing chemicals can form a corrosion cell that contributes to steel corrosion; and, bacteria and/or organism found in swaps or stagnant water can accelerate steel corrosion by producing sulfides at steel surfaces. "Steel construction errors",

including poor welds, poor fabricated detail quality, and excessive paint, can negatively affect the steel bridge elements' strengths and resistances to corrosion and thus lead to deficiencies.

- *Timber bridge element deficiency causes*: The most dominant causes of bridge timber deficiencies are "accidental loading", "fire", "insect", and "fungi". Fire consumes timber bridge elements at an extremely high rate (i.e., 0.05 inches per minute at the first 30 minutes), and thus could contribute to timber deficiencies (e.g., loss of section that reduces the loading-carrying capacity of timber elements) or even total consumption of a timber bridge. Because of the organic nature of timber, insects (e.g., termite and/or carpenter ant) could consume timber elements for shelters or foods, which would result in hollowed inside of timber elements and thus reduce the sections of timber elements. Fungi consumes timber in a similar fashion as insects. As reported, only "timber destructive fungi" (e.g., brown and white fungi) lead to timber deficiencies; while other types of fungi (e.g., mold and soft fungi) generally are not considered as destructive to timber.
- *Fiber reinforced polymer and masonry bridge element deficiency causes*: The main causes of bridge fiber reinforced polymer deficiencies include "extensive UV radiation" and "heat exposure". Bridge masonry deficiencies could be induced by causes, such as "frost-freeze", "marine growth", and/or "plant growth", etc. Detailed classifications of fiber reinforced polymer and masonry bridge element deficiency causes are presented in Figure 3.4.

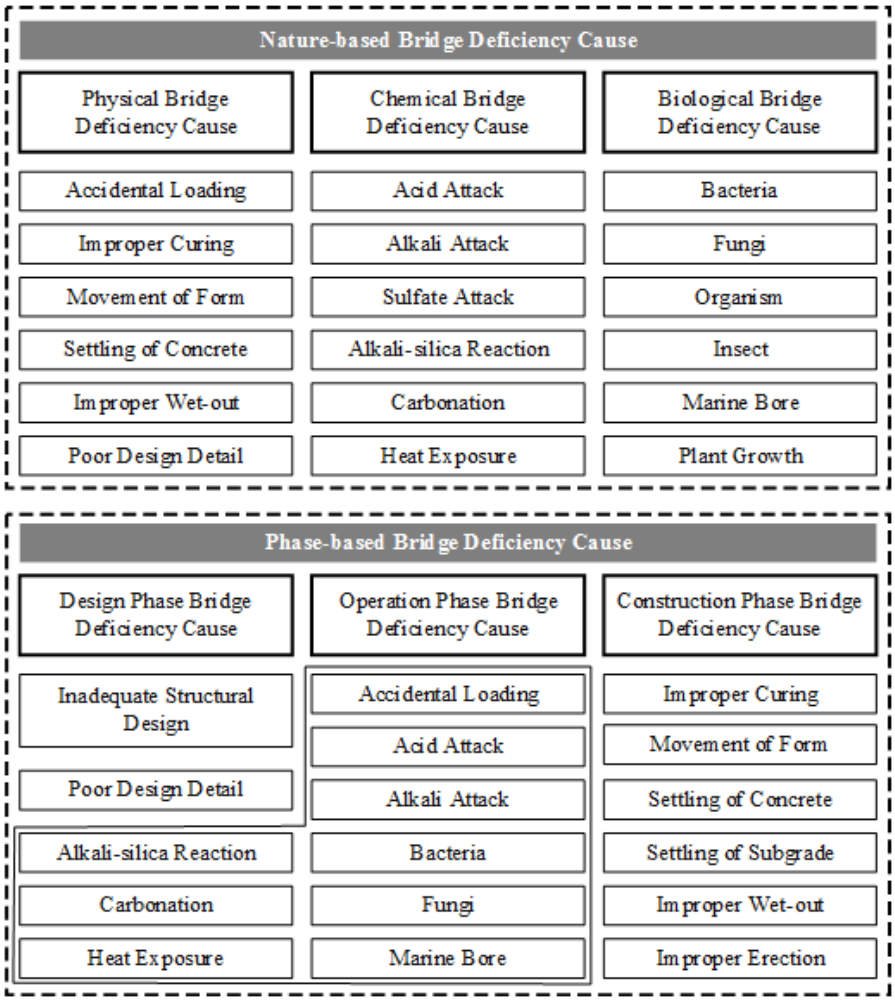


Figure 3.5. Bridge deficiency cause secondary modality view (partial).

3.1.2.4.2 *Secondary Modality Views*

The secondary modality views intend to represent bridge deficiency cause concepts from multiple perspectives for facilitating semantically-meaningful representations of the extracted information for supporting bridge performance understanding. All bridge deficiency cause concepts are reclassified based on: (1) the nature of the deficiency cause, and (2) the phase of the deficiency cause. The nature of deficiency cause is an intrinsic feature that characterizes a bridge deficiency cause. The phase of the deficiency cause characterizes bridge deficiency causes according to the different phases of a bridge's life cycle.

First, the “nature of deficiency cause” modality view classifies all bridge deficiency cause concepts into physical, chemical, and/or biological causes. A “physical bridge deficiency cause” is a cause that is characterized or produced by the forces and operational rules of physics, which could be an “accidental loading”, a “fire”, a “freeze-thaw”, a “freeze-frost”, a “moisture”, or an “overloading”. A “chemical bridge deficiency cause” is a cause that is related to or produced by chemical reactions, which could be a “chemical attack”, an “alkali-silica reaction”, or an “alkali attack”. A “biological bridge deficiency cause” is a cause that is related to or produced by living organisms or living processes, which could be a “bacteria”, “fungi”, “insect”, or “plant growth”, etc. Second, the “phase of deficiency cause” modality view categorizes all bridge deficiency cause concepts into a “design phase bridge deficiency design cause” (e.g., “inadequate structural design” and “poor design detail”), a “construction phase bridge deficiency design cause” (e.g., “improper consolidation” and “improper tooling”), or an “operation phase bridge deficiency design cause” (e.g., “frequent truck traffic” and “weathering”).

3.1.2.5 Bridge Maintenance Action Hierarchy

In the BridgeOnto, multimodality views for the bridge maintenance action hierarchy are presented. Different modality views of this hierarchy intend to capture the importance and context sensitiveness of bridge maintenance actions in bridge deterioration knowledge modeling. According to the AASHTO (2007), a “bridge maintenance action” could be a “corrective bridge maintenance action” or a “preventive bridge maintenance action”. A corrective bridge maintenance action is directed at repairing and/or replacing deteriorated bridge elements to reach their as-constructed or improved condition. A preventive bridge maintenance action is directed at preforming activities that will preserve bridge elements in their current or as-constructed condition and forestalling bridge deficiency developments. The classification of preventive bridge

maintenance action concepts is shown in Figure 3.6. The main and secondary modality views for corrective bridge maintenance action concepts are shown in Figure 3.6 and Figure 3.7. A partial list of the concepts with their corresponding sources is shown in Table 3.2.

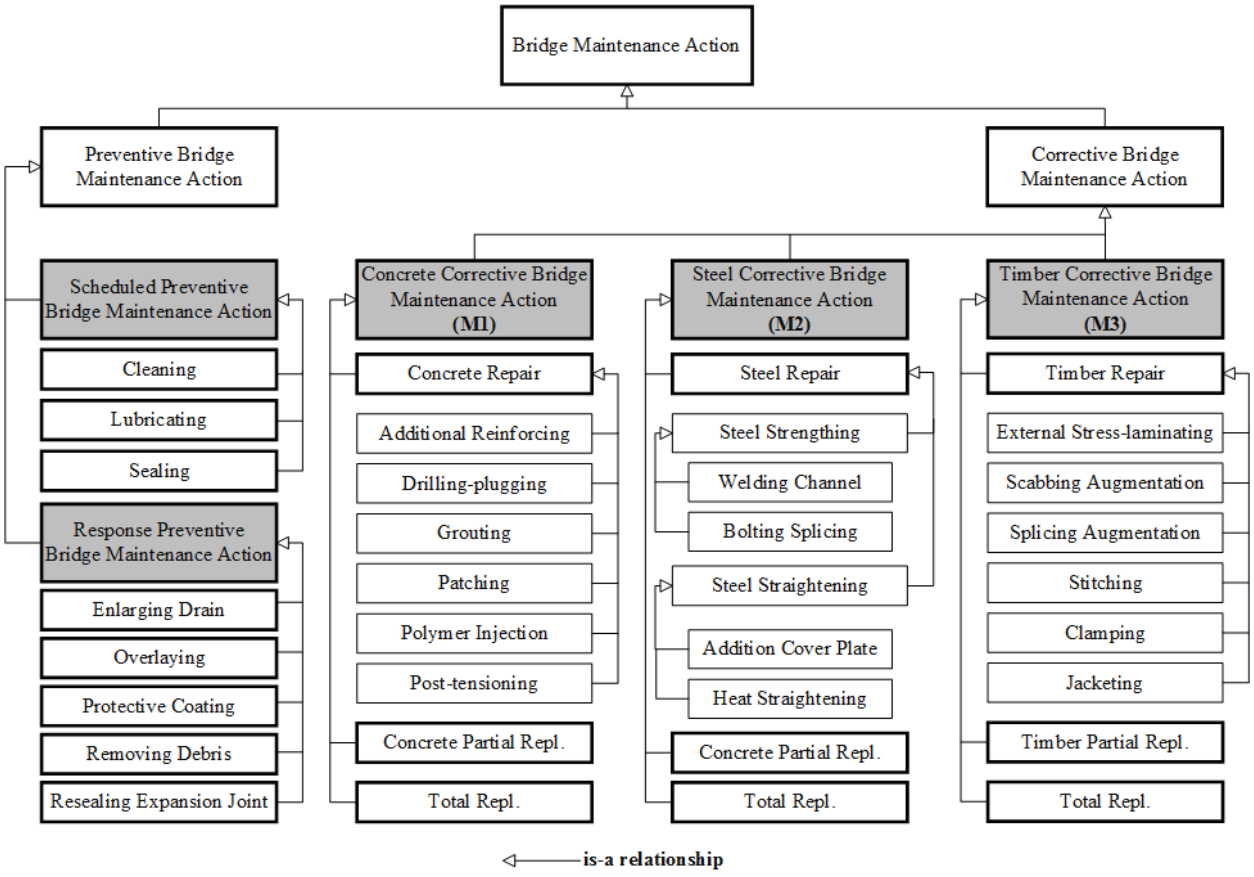


Figure 3.6. Bridge maintenance action hierarchy (partial).

3.1.2.5.1 *Main Modality View*

The subconcepts of the corrective bridge maintenance action hierarchy are further classified based on the most common bridge materials (i.e., concrete, steel, and timber, which collectively compose approximately 99% of the nation's bridges). In Figure 3.6, a "corrective bridge maintenance action" could be a "concrete bridge element corrective maintenance action", a "steel bridge element corrective maintenance action", or a "timber bridge element corrective maintenance action". For each type of corrective maintenance action, three primary subconcepts are defined:

- *Partial replacement*: the corrective bridge maintenance action that involves the partial removal of deteriorated sections from a bridge element and the partial placement of new sections.
- *Total replacement*: the corrective bridge maintenance action that involves the entire removal and replacement of a deteriorated bridge element whose deteriorated sections are typically over 50% of the entire bridge element.
- *Repair*: the corrective bridge maintenance action that does not involve replacement of bridge element section; instead, it utilizes different techniques to strength or restore a deteriorated bridge element.

For repair action, three main subconcepts are defined:

- *Concrete bridge element repair*: Concrete bridge element repair involves repairing concrete cracking to preclude moisture and chemicals entering the bridge elements (e.g., "additional reinforcement", "stitching", or "polymer injection"), repairing section loss to restore the load-carrying capacities of structural elements (e.g., "patching" or "jacketing"), and repairing prestress loss to rehabilitate the as-designed dynamic state of stress (e.g., "post-tensioning", or "reconnecting tendons with cable splicing"), etc.
- *Steel bridge element repair*: Steel bridge element repair involves repairing steel fatigue cracking to avoid sudden and catastrophic failure of steel elements (e.g., "crack arresting hole"), strengthening steel elements to recover or improve their load-carrying capacity (e.g., "welding plate" or "bolting channel"), and straightening deformed steel elements to address damage from impacts or collisions (e.g., "heat straightening" or "introduction of composite action"), etc.
- *Timber bridge element repair*: Because timber bridge elements are easy and economical to be maintained by replacement, timber repair mainly focuses on cracking arresting actions (e.g.,

"clamping", "scabbing augmentation", or "splicing augmentation") when a timber replacement is unfeasible.

3.1.2.5.2 *Secondary Modality Views*

Figure 3.7 shows a partial view for the secondary modality views of the physical bridge maintenance action hierarchy. The bridge maintenance knowledge is multifaceted and context sensitive, because it involves complex decision making to select the most reliable physical maintenance actions under various context constrains, such as maintenance cost, availability of crew, material, and equipment, and type of bridge element, etc. To facilitate the representation of bridge maintenance knowledge, multiple secondary modality views for the physical bridge maintenance actions are defined, based on: agency, environmental restriction, maintenance bridge element, practice attribute, and implementation context.

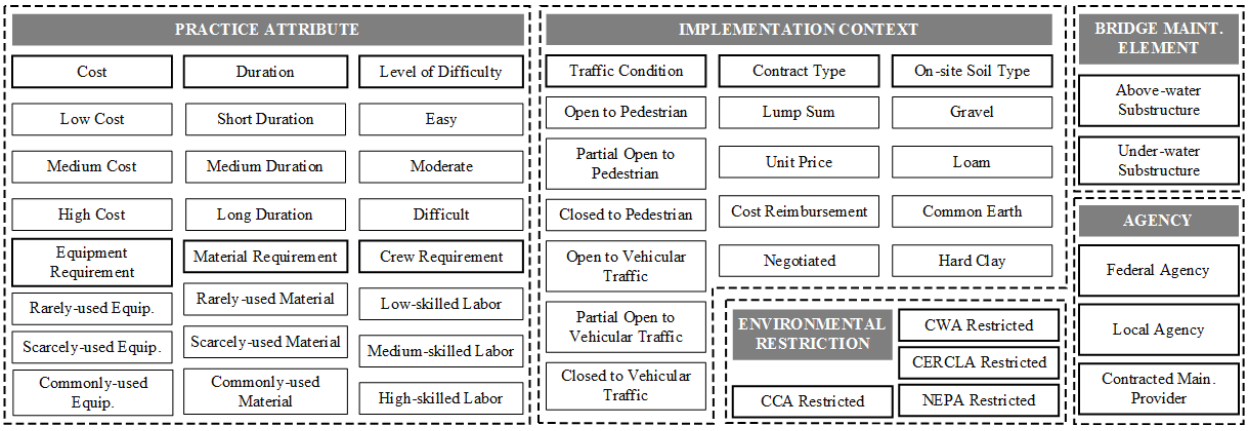


Figure 3.7. Bridge maintenance action secondary modality view (partial).

As presented in Figure 3.7, an "agency" is defined as a group or an organization that leads corrective maintenance actions, which could be a "federal agency", a "local agency", or a "contracted maintenance provider". An "environmental restriction" refers to an environmental regulation that could affect the feasibility of a corrective bridge maintenance action, which could

include the Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA), the Clean Air Act (CAA), the Clean Water Act (CWA), or the National Environmental Policy Act (NEPA), etc. An environmental restriction could affect the applicability of specific maintenance actions, and thus could determine the different classifications of maintenance actions based on restrictions. For example, maintenance actions involving air blasting are subject to the CAA airborne lead emission requirement. A "maintenance bridge element" refers to the bridge elements that characterize different corrective bridge maintenance actions, which could be a "deck", a "beam", a "truss", an "above-water substructure", an "under-water substructure", or a "pile", etc. Based on the "maintenance bridge element" modality view, a corrective bridge maintenance action could be characterized accordingly as a "deck corrective bridge maintenance action", an "above-water substructure corrective bridge maintenance action", or an "under-water corrective bridge maintenance action", etc. A "practice attribute" refers to a property or characteristic that defines or describes a physical bridge maintenance action. The practice attribute modality emphasizes six attributes that are critical in selecting suitable corrective bridge maintenance actions: cost, duration, level of difficulty, equipment requirement, crew requirement, and material requirement. The six attributes correspond to the "bridge maintenance process attributes" defined in the bridge attribute hierarchy. To characterize each of these six attributes, three scales are defined: low, medium, and high. For example, a corrective bridge maintenance action could be classified as a low-material-requirement action, a medium-material-requirement action, or a high-material-requirement action. The "implementation context" refers to the circumstance in which a corrective bridge maintenance action is being implemented, which could be a "traffic condition", an "on-site soil type", or a "contract type", etc. For example, a corrective

bridge maintenance action could be classified as an open-to-pedestrian-traffic action, an open-to-vehicular-traffic action, or a closed-to-vehicular-traffic action.

3.1.2.6 Bridge Attribute Hierarchy

A bridge attribute is defined as a characteristic (e.g., a bridge element material or a bridge inspection method) that describes a "thing" (e.g., a bridge element or a bridge deficiency). In the BridgeOnto, a "bridge attribute", as shown in Figure 3.8, is described by three subconcepts:

- *Bridge element attribute*: A "bridge element attribute" is identified as a "bridge element material" (e.g., concrete or steel) or a "bridge element configuration" (i.e., the constructed shape of a bridge element).
- *Bridge process attribute*: A "bridge process attribute" could be a "bridge inspection process attribute" or a "bridge maintenance process attribute". A "bridge inspection process attribute" describes the contextual characteristics of a "bridge deficiency" in two primary dimensions – a "bridge inspection date" and a "bridge inspection method". The "bridge inspect date" reflects the deficiency discovery date in order to capture how a bridge deficiency is propagating over time; and, the "bridge inspection method" affects the confidence levels of a recorded deficiency measurement. For example, deficiency measurement information from physical inspection methods is generally more accurate than that from visual inspection methods. The "bridge maintenance process attribute" represents the contextual characteristics of a "bridge maintenance action" in eight primary dimensions, as shown in Figure 3.8. The "bridge maintenance material" is one of the most important contextual dimensions in defining a maintenance action, because it could be positively or negatively affect the performance of an action. For example, asphalt material for concrete deck patching is considered as a temporary repair rather than a permanent one. In order to describe this important knowledge about how a

deficiency is maintained, it is necessary to model the different types of materials that are commonly used in bridge maintenance practices. In the BridgeOnto, a "bridge maintenance material" could be a "sealant" (e.g., siloxane or methyl methacrylate), a "cementitious material" (e.g., silica fume concrete or fiber reinforced concrete), a "paint" (e.g., epoxy paint or zinc-rich primer), or a "wood preservative" (e.g., alkaline copper quaternary or oxine copper), etc.

- *Bridge deficiency attribute*: A "bridge deficiency attribute" could be a "bridge deficiency onset date" or a "bridge deficiency measurement". The "deficiency onset date" could be same or different from the bridge inspection date, depending on how bridge inspectors record them. The "bridge deficiency measurement" could be a "numerical geometry measurement" associated with a "numerical geometry measurement unit" (e.g., the numerical measurement of length in inch), a "categorical severity measurement", or a "categorical quantity measurement". These measurements do not mutually exclude each other, because they define orthogonal measurement dimensions of a deficiency.

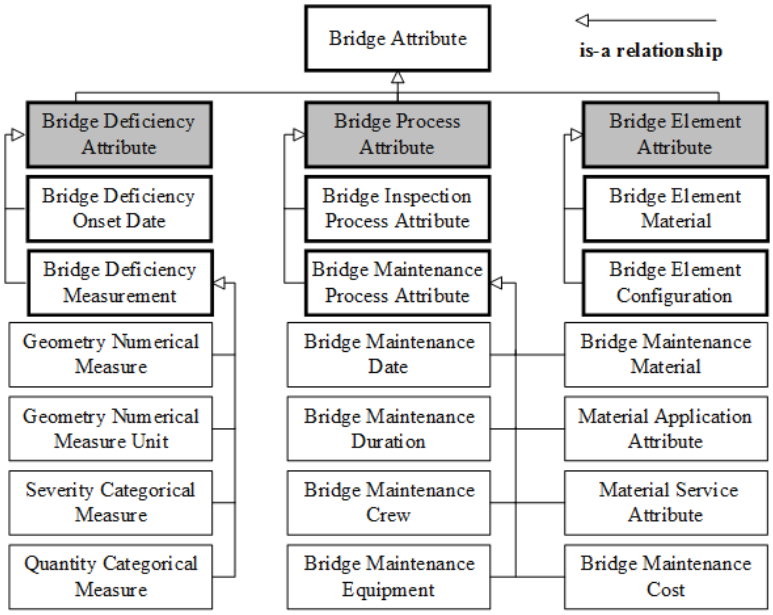


Figure 3.8. Bridge attribute hierarchy (partial).

3.1.3 BridgeOnto Coding

The BridgeOnto was coded using Protégé 3.4.5 (Protégé 2016). Protégé is an off-the-shelf ontology editor that supports coding ontology in the Web Ontology Language (OWL) format. The BridgeOnto coding included the following two main steps. First, the concepts were represented as Protégé-OWL classes and the taxonomy of the concepts was captured and coded into a superclass-subclass hierarchy. Second, the relations between the concepts were represented and coded using Protégé-OWL "extension property restrictions" and "necessary conditions".

3.2 BridgeOnto Evaluation

The evaluation of the BridgeOnto included two primary steps: verification and validation. Ontology verification aimed to ensure that the ontology was constructed correctly and consistently towards implementing its defined requirements, which included ensuring that the ontology satisfies its functional requirements and ensuring that the ontology is free of redundancy and inconsistency errors. Ontology validation aimed to evaluate the capability of the ontology in modeling the real-world that it tries to model. The BridgeOnto verification and validation methods included: (1) answering CQs, (2) automated consistency and redundancy checking, (3) expert interviews, and (4) application-oriented validation. These methods, collectively, intended to evaluate whether or not the BridgeOnto meets the following criteria (Gómez-Pérez et al. 2006; Gruber 1995; Vrandečić 2009). The methods and the corresponding criteria are presented in Table 3.3.

- *Clarity*: Does the terms in the BridgeOnto communicate the intended meaning of the concepts and relations clearly?
- *Representation*: How representative are the concepts of the BridgeOnto?

- *Coverage*: Does the BridgeOnto cover the main concepts and relationships in its domain of interest?
- *Conciseness*: Is the BridgeOnto free of unnecessary and redundant concepts and relationships?
- *Consistency:* Are the representations of the BridgeOnto consistent?
- *Navigational Ease*: Is the BridgeOnto easy to navigate?
- *Extendibility*: Does the BridgeOnto support the extension to cover other domains of interest (i.e., inclusion of additional concepts and relationships to extend its domain of interest)?
- *Applicability:* How applicable can the BridgeOnto be in specific application scenarios or user cases?

Table 3.3. BridgeOnto evaluation methods and evaluation criteria.

Evaluation method	Criterion							
	1	2	3	4	5	6	7	8
Answering competency questions	O	X	O	O	O	O	O	O
Automated consistency and redundancy checking	O	O	O	X	X	O	O	O
Expert interview	X	X	X	X	O	X	X	O
Application-oriented validation	O	O	O	O	O	X	O	X

1 – 8: Clarity, representation, coverage, conciseness, consistency, navigational ease, extendibility, applicability; X = Evaluate; O = Do not evaluate.

3.2.1 BridgeOnto Verification

3.2.1.1 Answering Competency Questions

CQs have been commonly utilized as guidance to develop an ontology and/or as functional requirements to verify an ontology. An example of a CQ used in the BridgeOnto verification process is "*what are the deficiency types that a timber deck could have*?" The ability of the ontology to answer all CQs was checked. It was shown that the BridgeOnto is able to answer all CQs.

3.2.1.2 Automated Consistency and Redundancy Checking

The ontology was automatically checked for consistency and redundancy errors using the Pellet 1.5.2 Reasoner in Protégé 3.4.5 (Protégé 2016). The defined concepts and relationships of the BridgeOnto passed the consistency and redundancy checks.

3.2.2 BridgeOnto Validation: Expert Interview Validation

Benchmarking the methodology by El-Gohary and El-Diraby (2010), human expert interviews were conducted to solicit expert feedback about the capability of the BridgeOnto in modeling bridge deterioration knowledge. A purposive sampling method was used to select the expert interview participants. Purposive sampling is a non-probabilistic sampling method that is especially effective when a certain type of participants such as knowledgeable experts need to be recruited for a study (Tongco 2007). The underlying philosophy of purposive sampling is that participants are selected based on a set of predetermined criteria meeting a particular research objective (Guest et al. 2006). Therefore, to select participants for the expert interviews, three criteria were defined: (1) familiarity with the bridge domain, (2) in-depth understanding of bridge deterioration mechanisms and bridge maintenance practices, and (3) awareness of information/knowledge modeling. Based on these criteria, a total of eight expert participants from both academia and industry were selected. They have an average of 17.6 years of experience in bridge-related areas, including bridge design, inspection, and maintenance, bridge and structural engineering, and structure repairing and retrofitting. The details of the background of the experts are summarized in Table 3.4.

Table 3.4. Expert interview participant background.

Respondent	Field of experience	Years of experience	Organization type
1	Highway bridge design and bridge inspection	4	Industrial
2	Bridge design, bridge inspection contract, and bridge inspection training	43	Industrial
3	Bridge maintenance	29	Industrial
4	Bridge inspection	3	Industrial
5	Structural engineering	30	Academic
6	Bridge engineering and construction engineering and management	10	Academic
7	Structure repairing and retrofitting	2	Academic
8	Bridge engineering and structure retrofitting	20	Academic

Each expert interview session was comprised of three parts: (1) a brief introduction to the purpose and scope of the research, the high-level concepts of the BridgeOnto, and the application scenario of the ontology; (2) a detailed introduction to the ontology (including concepts and relationships) using the ontology tree presentation by Protégé 3.4.5 (Protégé 2016). In this part, the participant could also navigate through the ontology tree himself/herself; and (3) a structured survey of the participant's opinion using a questionnaire. The survey aimed to solicit expert feedback on the following main ontology evaluation criteria: clarity, representation, coverage, conciseness, navigational ease, and extendibility.

The questionnaire is composed of the following eight sections:

1. Participant information: This section aimed to solicit some background information about the participants, including organization, position, year of experience, and contact information.
2. Background and familiarity with survey scope: This section aimed to confirm that the participants meet the aforementioned participant selection criteria. In this section, the participant's level of familiarity with bridge deterioration mechanisms and bridge

inspection/maintenance practices and level of awareness of information/knowledge modeling were assessed by two direct questions.

3. Need for the BridgeOnto: This section aimed to solicit expert opinion on the need for bridge document analytics and the need for bridge deterioration knowledge modeling for supporting such analytics by two direct questions.
4. Term clarify: This section aimed assess the clarity of the terms in effectively communicating the intended meaning of the concepts. In this section, participants evaluated the term clarity of a set of randomly pre-selected concepts.
5. Classification: This section aimed to evaluate the abstraction and categorization effectiveness of the classifications. In this section, the participants were requested to rate their levels of agreement with the hierarchical paths of a set of concepts.
6. Navigational ease: In this section, participants were requested to navigate through the ontology to find a set of randomly-selected concepts in the taxonomy, and then rate the level of ease to find each concept.
7. Conciseness and extendibility: This section asked participant whether they found unnecessary or redundant concepts and whether they think the BridgeOnto can be extended to other civil infrastructure subdomains.
8. Overall assessment: This section aimed to seek an overall assessment of the BridgeOnto through six direct questions about the clarity, representation, coverage, conciseness, navigational ease, and classification of the ontology.

The interview results for Sections 2-8 of the questionnaire are summarized in Table 3.5, where the responses to the conciseness and extendibility criteria were recorded using a binary scale (with 1 and 0 standing for “yes” and “no”, respectively) and the responses to the other criteria were

recorded using a six-point Likert scale (with 1 and 6 standing for the most and the least favorable, respectively). Overall, the participants collectively "agree" with the classification of the BridgeOnto, and found the ontology "very ease" to navigate, the concepts "familiar", "very representative", and "very concise", and the terms "effectively" communicating the intended meaning of the concepts. The participants also collectively indicated that the concepts and relations of the BridgeOnto "cover" the bridge deterioration knowledge aspects. The overall opinion of the participant also indicate that the BridgeOnto can be extended to other civil infrastructure subdomains, such as highway and roadway maintenance, building maintenance, dam structure maintenance, etc.

Table 3.5. Summary of expert interview results.

Section	Question	Mean	Median	Standard deviation	Interpretation of result (based on median)
2[a]	**Background and familiarity**				
	How familiar are you with bridge deterioration mechanisms and bridge inspection/maintenance practices?	2.13	2.00	0.64	Familiar
	To what extent are you aware of information/knowledge modeling?	2.75	3.00	0.71	Somewhat aware
3[a]	**Need for bridge document analytics and bridge deterioration knowledge modeling**				
	Do you think bridge document analytics are important to bridge management?	1.50	1.00	0.53	Very important
	Do you think bridge deterioration knowledge modeling is important to the success of bridge document analytics?	1.75	2.00	0.71	Important
4[a]	**Term clarity**	1.92	2.00	0.90	Clear
5[a]	**Classification**	1.75	2.00	0.69	Agree
6[a]	**Navigational ease**	1.48	1.00	0.91	Very easy
7[b]	**Conciseness and extendibility**				
	Do you find any unnecessary or useless concepts?	0.00	0.00	0.00	No
	Do you find any redundant concepts?	0.13	0.00	0.35	No
	Do you think the BridgeOnto could be extended to represent the knowledge of other civil infrastructure sub-domains?	0.75	1.00	0.46	Yes
8[a]	**Overall assessment**				
	Do you agree with the main classification of the BridgeOnto?	1.75	2.00	0.71	Agree
	How easy was it to navigate through the BridgeOnto?	1.50	1.00	0.76	Very easy
	How familiar are the concepts of the BridgeOnto?	2.25	2.00	1.04	Familiar
	How representative are the concepts of the BridgeOnto?	1.63	1.00	0.74	Very representative
	what do you think of the conciseness of the BridgeOnto?	1.63	1.00	0.74	Very Concise
	How effectively do you think the terms used in the BridgeOnto communicate the intended meaning?	2.00	2.00	0.76	Effectively
	Overall, do you think the BridgeOnto cover the main concepts and relations of the defined bridge deterioration knowledge aspects?	1.75	2.00	0.71	Complete

[a] Six-point Likert scale;
[b] Binary scale.

CHAPTER 4 – SEMANTIC INFORMATION EXTRACTION

This chapter presents the proposed information extraction method for extracting information about bridge conditions and maintenance actions from textual bridge inspection reports. The method development and evaluation (Research Task #3) are presented in this chapter.

4.1 Comparison to the State of the Art

Automated IE from bridge inspection reports – compared to other IE efforts such as IE from building codes (e.g., Zhang and El-Gohary 2013) and social media (e.g., Ritter et al. 2012) – is challenging because of two main reasons. First, bridge inspection reports are highly variable in terms of text characteristics and patterns, because they are typically written by many different writers/inspectors from various local, state, and federal agencies. Existing rule-based (e.g., Appelt et al. 1993; Elsebai et al. 2009; Lehnert et al. 1991; Riloff 1993; Xu et al. 2010) and supervised machine learning (ML)-based (Li et al. 2013) IE methods would, thus, require an unaffordable amount of human effort for developing a comprehensive set of representative pattern-matching-based rules or annotated training data, in order to capture the variability in text patterns. Second, on one hand, bridge inspection reports exhibit domain-specific uniqueness that involves complex concept identification and relationship association (i.e., identifying complex technical concepts about bridge elements, deficiencies, and maintenance actions, etc., and their associated relations). On the other hand, because of the technical criticality of the extracted data/information, a high performance in both precision and recall is required for the automated IE from bridge inspection reports. Existing semi-supervised ML-based (e.g., Guo et al. 2009; Jiang and Zhai 2007; Liao and Veeramachaneni 2009; Liu et al. 2011) and unsupervised ML-based (e.g., Alfonseca and Manandhar 2002; Etzioni et al. 2005; Nadeau et al. 2006; Shinyama and Sekine 2004) IE methods cannot deal such complexities and variabilities with a high precision and recall performance.

4.2 Information Extraction Method Development

4.2.1 Proposed Information Extraction Method

To address the above-mentioned knowledge gaps, a new ontology-based, semi-supervised conditional random fields (CRF)-based information extraction (IE) method is proposed. The proposed IE method allows for capturing the dependency structures as well as the distributions of both labeled and unlabeled data simultaneously in a concave machine-learning function. It, thus, dynamically adapts itself to unseen instances by further learning from a large number of unlabeled data – in addition to learning from a small set of fixed labeled data – to save human effort and achieve a high IE performance. The problem of extracting information from bridge inspection reports is defined as an automated named entity recognition and classification (NERC) task. In this thesis, the NERC task aims to automatically recognize and classify information units (i.e., named entities) into predefined entity classes. As explained in Section 1.5.2.1, the entity classes were predefined based on the analyses of sample bridge inspection reports, from both bridge engineering and NLP perspectives. The defined entity classes (i.e., target information types) include: bridge element, deficiency, deficiency cause, maintenance action, maintenance material, numerical measure, numerical measure unit, categorical quantity measure, categorical severity measure, date, and other.

The proposed IE method is novel in the following primary ways:

1. Capturing the dependency structures as well as the distributions of a small set of fixed labeled data and a large set of unlabeled data simultaneously in a semi-supervised CRF-based machine-learning function. Two types of dependency structures of the data are explicitly defined and represented: the dependencies between entity classes and the dependencies between entities

(i.e., a word such as “deck”) and their classes (i.e., its corresponding class such as “bridge element”). The distributions of the data are captured under the semi-supervised learning cluster assumption to enable the utilization of a large set of unlabeled data with their derived entity class sequences, in order to dynamically adapt to unseen instances. Modeled in this way, the proposed method is expected to save human-annotation effort and achieve high recall and precision.

2. Formulating the NERC task into a concave semi-supervised machine-learning function. The proposed function is composed of two primary components, one for labeled data and one for unlabeled data, that both follow the exponential family distribution in logarithmic space. Because the log-likelihood of exponential families is strictly concave, the linear combination of two such distributions also holds for concavity. The concavity of the proposed method is extremely important to avoid suboptimal initializations and converging at local maxima, which would otherwise negatively affect IE performance.
3. Utilizing formally defined semantics for assisting IE. The semantics defined by the BridgeOnto ontology (presented in Chapter 3), are utilized to facilitate IE from bridge inspection reports based on content and domain-specific meaning. Each word in the reports is compared to the concepts in the BridgeOnto, and is mapped to the highest-level classification of a concept if the concept contains the word. In this way, the words are semantically represented to avoid ambiguities of word senses, thereby improving IE performance.

Figure 4.1 depicts the high-level algorithm for the proposed IE method. In the following subsections, the proposed IE method and its core components are presented in more detail.

Algorithm: Ontology-Based, Semi-Supervised CRF-Based Information Extraction

Input labeled data
Output labeled data: $L = \{(\mathbf{x}_i, \mathbf{y}_i)\}, i = 1, \ldots, l$
For each one of raw textual bridge inspection reports
    Execute text preprocessing and feature extraction
    Output unlabeled data: $U = \{(\mathbf{x}_i)\}, i = l+1, \ldots, l+u$
    Construct heterogeneous information network (G)
    Define meta-paths (P)
    For each token t_1 in $L + U$
        For each token t_2 in $L + U$
            For each P in G
                Compute semantic similarity $S(t_1, t_2)$ as per Eq. (7)
    For $i = l+1 \rightarrow l+u$
        Derive entity class sequence $\tilde{\mathbf{y}}_i$ for $\mathbf{x}_i$ based on semantic similarity
    Output $U = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}, i = l+1, \ldots, l+u$
    Define $\sum_{i=1}^{l+u} \log P(\mathbf{y}_i|\mathbf{x}_i)$ as per Eq. (1)
    Repeat
        Train $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)/(\mathbf{x}_i, \mathbf{y}_i)$ $(1 \le i \le l+u)$ with limited-memory BFGS
    Until converged
    Output $\theta = \{\lambda_k\}_{k=1}^{K}$
    For $i = l+1 \rightarrow l+u$
        $\mathbf{y}_i = \underset{\mathbf{y}_i \in \mathbf{y}_i'}{\operatorname{argmax}} \log P(\mathbf{y}_i|\mathbf{x}_i; \theta)$
    Output predicted entity class sequences $\{(\mathbf{y}_i)\}, i = l+1, \ldots, l+u$

Figure 4.1. High-level algorithm for the proposed IE method.

4.2.1.1 Proposed Information Extraction Model

The proposed ontology-based, semi-supervised CRF-based IE model is the backbone of the proposed IE method. In this chapter, a dataset is defined as $D = L \cup U$, where D consists of l labeled data, $L = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$, and u unlabeled data, $U = \{(\mathbf{x}_i)\}_{i=l+1}^{l+u}$. L is intended to be a fixed (i.e., same set used for all IE tasks) labeled dataset, containing labeled sentences from a selected training text/report, for training the proposed algorithm. U is an unlabeled dataset, containing unlabeled sentences from the report from which information needs to be extracted, which is used to dynamically adapt the algorithm to the unseen instances (i.e., to the unseen text from which the information needs to be extracted during an IE task). The unlabeled dataset, thus, varies from an

IE task to another. $\mathbf{x}_i$ denotes the i^{th} structured observation (i.e., a preprocessed natural language sentence in its feature representation). $\mathbf{y}_i$ denotes the i^{th} structured output (i.e., a sequence of entity classes each of which corresponds to a token in the $\mathbf{x}_i$). The goal of the proposed IE model is to learn from both labeled and unlabeled data to fine-tune the model weights (i.e., λ_k) for maximizing $\sum_{i=1}^{l+u} \log P(\mathbf{y}_i|\mathbf{x}_i)$, such that the learned model can predict a structured output $\mathbf{y}_i$ $(l+1 \leq i \leq l+u)$ for each unlabeled data accurately. $\log P(\mathbf{y}_i|\mathbf{x}_i)$, as defined by supervised CRF (Lafferty et al. 2001), is the conditional probability of $\mathbf{y}_i$ given $\mathbf{x}_i$ in logarithmic space. Supervised CRF only aims to maximize $\sum_{i=1}^{l} \log P(\mathbf{y}_i|\mathbf{x}_i)$ with L, without considering the dependency structures and distributions of U. The proposed IE model is built on supervised CRF, but makes significant improvements so as to – in addition to learning from a small set of fixed labeled data – dynamically adapt itself to unseen instances by further learning from a large collection of unlabeled data, to reduce human effort and achieve a high IE performance. The proposed semi-supervised CRF model is defined in Eq. (4.1) as follows:

$$\sum_{i=1}^{l+u} log\, P(\mathbf{y}_i|\mathbf{x}_i) = LI + UI - \gamma \sum_{k=1}^{K} {\lambda_k}^2 \tag{4.1}$$

In Eq. (4.1), the labeled item (i.e., LI) is the mathematical representation of supervised linear-chain CRF for labeled data (Lafferty et al. 2001). The supervised linear-chain CRF model, which is a special case to the general supervised CRF, assumes that the entity class of the current token depends on the features defined by the current token and its preceding entity class. The labeled item in Eq. (4.1) is defined in Eq. (4.2), where l denotes the number of labeled data (i.e., preprocessed natural language sentences in feature representations), T denotes the length of the i^{th} $(1 \leq i \leq l)$ labeled data, and K denotes the number of feature functions. $f_k(\mathbf{y}_i^t, \mathbf{y}_i^{t-1}, r, m)$ in

Eq. (4.2) is the k^{th} feature function with a model weight of λ_k. A detailed introduction to the feature function is presented in Section 4.2.1.2.

$$LI = \sum_{i=1}^{l} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\boldsymbol{y}_i^t, \boldsymbol{y}_i^{t-1}, r, m) - \sum_{i=1}^{l} log\, Z(\mathbf{x}_i) \tag{4.2}$$

In Eq. (4.1), the unlabeled item (i.e., UI) is the mathematical representation of the proposed semi-supervised linear-chain CRF for unlabeled data. This item serves as an adjustment item to the supervised item (i.e., LI). The unlabeled item is proposed under the semi-supervised learning cluster assumption, which states that if two data points lay in the same cluster, then they are likely to have a similar class label (Mann and McCallum 2007). In the absence of labeled entity classes for unlabeled data, this item tries to (1) derive entity class sequences for each unlabeled data based on the entity class sequences of its similar data that were labeled, and (2) learn from unlabeled data and their derived entity class sequences to dynamically adjust the proposed IE model, such that to adapt itself to unseen instances. To achieve these goals, the unlabeled item is mathematically defined in Eq. (4.3), where u denotes the number of unlabeled data (i.e., preprocessed natural language sentences in feature representations), N denotes the number of the most similar entity class sequences (i.e., derived entity class sequences, $\tilde{\mathbf{y}}_i$) for the i^{th} $(l+1 \leq i \leq l+u)$ unlabeled data, and G denotes a heterogeneous information network constructed for computing token-to-token semantic similarities between labeled and unlabeled tokens. $P(\tilde{\mathbf{y}}_{i,n}|\mathbf{x}_i, G)$ denotes the likelihood of the n^{th} derived entity class sequences given $\mathbf{x}_i$ $(l+1 \leq i \leq l+u)$ and G, and is used to parameterize the importance of each derived entity class sequence according to its similarity degree. Other parameters in Eq. (4.3) follow those defined in Eq. (4.2). The methods used for constructing the heterogeneous information network, for computing semantic similarities

to derive $\tilde{\mathbf{y}}_i$ $(l+1 \leq i \leq l+u)$, and for computing $P(\tilde{\mathbf{y}}_{i,n}|\mathbf{x}_i, G)$ $(l+1 \leq i \leq l+u)$ are further introduced in Section 4.2.1.3.

$$UI = \left\{ \sum_{i=l+1}^{l+u} \sum_{n=1}^{N} P(\tilde{\mathbf{y}}_{i,n}|\boldsymbol{x}_i, G) \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\tilde{\mathbf{y}}_{i,n}^{t}, \tilde{\mathbf{y}}_{i,n}^{t-1}, r, m) \right\} - \sum_{i=l+1}^{l+u} \log Z(\mathbf{x}_i) \tag{4.3}$$

In Eqs. (4.2) and (4.3), $\sum_{i=1}^{l} \log Z(\mathbf{x}_i)$ and $\sum_{i=l+1}^{l+u} \log Z(\mathbf{x}_i)$ are normalization constants in logarithmic space to the LI and UI, respectively. The normalization constant is the sum of all possible entity class sequences (i.e., $\mathbf{y}_i'$) for a given $\mathbf{x}_i$ $(1 \leq i \leq l+u)$. It is needed to guarantee that the $\sum_{\mathbf{y}_i'} P(\mathbf{y}_i|\mathbf{x}_i)$ $(1 \leq i \leq l+u)$ equals to one. The normalization constant $Z(\mathbf{x}_i)$ is defined in Eq. (4.4), where $\mathbf{y}_i'$ denotes all possible entity class sequences for a given $\mathbf{x}_i$ $(1 \leq i \leq l+u)$. Other parameters in Eq. (4.4) follow those defined in Eqs. (4.2) and (4.3).

$$Z(\mathbf{x}_i) = \sum\nolimits_{\mathbf{y}_i'} exp\left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{y}_i^t, \mathbf{y}_i^{t-1}, r, m) \right) \tag{4.4}$$

In Eq. (4.1), γ is a regularization item weight that penalizes each model weight (i.e., λ_k) to control the degree of overfitting or underfitting. γ, thus, needs to be fine-tuned to avoid overfitting or underfitting. The empirical results on fine-tuning γ are presented in Section 4.3.2.4.

4.2.1.2 Features and Feature Function

The proposed IE method models a set of interdependent text features for facilitating IE from bridge inspection reports. A new feature representation is proposed, with both syntaxes and formally defined semantics in a context window of size one, to represent each token in a sentence. The context window is constructed with the features of the current token, as well as the features of the preceding and succeeding tokens. It intends to provide information on how the current token

should be interpreted and classified based on the features of the surrounding tokens, not only those of the current token. As shown in Figure 4.2, the proposed feature representation is a 1×36 feature vector, including both syntactic and semantic features. The syntactic features include original tokens, stems, and part-of-speech (POS) tags. A stem is the root form of a token without inflectional and derivational suffixes and affixes. For example, "paint" is the stem of "repainted". A POS tag describes the syntactic word class (also referred as lexical category) of a token based on its context in a sentence. In this thesis, the syntactic word classes by the Penn Treebank were used (Marcus et al. 1993), which include determiner, adjective, noun, and verb, etc. The semantic features were recognized and extracted based on the BridgeOnto (developed as per Research Task #2, in Chapter 3) by comparing the token's stem to the stem(s) of each concept in the ontology.

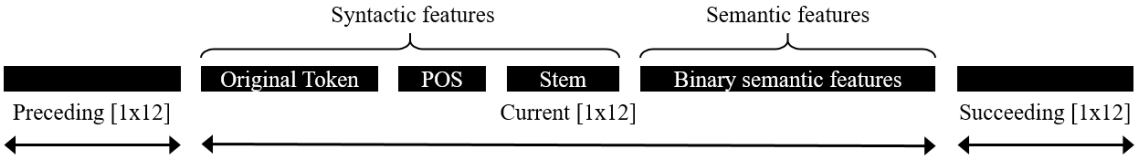


Figure 4.2. Proposed feature representation.

A feature function is an indicator function that indicates whether or not the current token (i.e., $\mathbf{x}_i^t$) can be labeled with the current entity class (i.e., $\mathbf{y}_i^t$). Formally, a feature function [i.e., $f_k(\mathbf{y}_i^t, \mathbf{y}_i^{t-1}, r, m)$] is defined with an input template that contains: the entity classes of the current token and the preceding token (i.e., $\mathbf{y}_i^t$ and $\mathbf{y}_i^{t-1}$), a feature (i.e., r) from the defined feature space, and the position (i.e., m) of the current token's feature vector. If the current token carries the exact feature at the exact position in its feature vector as defined in the input template, and its preceding token has an entity class as defined in the input template, a feature function returns 1; otherwise, it returns 0. The returned value of 1 indicates that the current token could be labeled with the current entity class based on this feature function. An example of a feature function is presented

in Eq. (4.5), where "ET" denotes the entity class for "bridge element", and "NN" denotes the POS tag feature for "noun". Given this feature function, a token could be labeled as "ET", if its preceding token's entity class is "ET" and this token's 15th feature is "NN".

$$f_1(\boldsymbol{y}_i^t = "ET",\ \boldsymbol{y}_i^{t-1} = "ET",\ "NN", 15) = \begin{cases} 1 & if\ the\ t^{th}\ token\ carries\ the\ input \\ 0 & otherwise \end{cases} \quad (4.5)$$

Because of its rich features and multiple entity classes, the proposed IE model utilizes a set of feature functions to collectively determine the confidence level of a token being labeled with a specific entity class. The number of feature functions [i.e., K in Eq. (4.1)] is determined by the number of entity classes and the number of distinctive features in the defined feature space. Thus, the number of feature functions $K = |entity\ class|^2 \times |distinctive\ feature|$. In addition, because certain feature and entity class combinations are more informative, each feature function f_k is parameterized by a model weight λ_k.

4.2.1.3 Semantic Similarity Measure

A semantic similarity (SS) measure is needed to derive the most likely entity class sequences [i.e., $\tilde{\boldsymbol{y}}$ in Eq. (4.3)] for unlabeled data. Then, the proposed IE model can learn from both labeled and unlabeled data (with derived entity class sequences) for a high IE performance. The most likely entity class sequences for unlabeled data are derived based on the entity classes of labeled data that are semantically similar to unlabeled data. In this thesis, it is required that the SS measure: (1) considers both types of SS, corpus-based SS (i.e., similarity degree of a token pair based on the distributions of their syntactic features in the dataset) and knowledge-based SS (i.e., similarity degree of a token pair based on the distributions of their semantic features in the dataset); (2) considers intra-class concepts as being semantically identical (e.g., "deck" and "truss" are semantically same under the "bridge element" entity class); and (3) captures the context in which

the subject token exists (i.e., considering the semantics defined by the preceding and succeeding tokens of the subject token) to represent and capture the polymorphic meanings of natural language (e.g., "deck" in "concrete deck overlay" and in "asphalt decking repair"). According to these requirements, following Sun et al. (2011), this thesis first applies the PathSim in a heterogeneous information network to compute token-to-token semantic similarities (SSs). Then, this thesis proposes to linearly combine different types of token-to-token SSs that are defined by the meta-paths. Finally, the thesis proposes to recover the top N similar entity class sequences and to compute the normalized likelihood [i.e., $P(\tilde{\mathbf{y}}_{i,n}|\mathbf{x}_i, G)$] for each derived entity class sequence, by assuming independencies between tokens in each unlabeled data/sentence.

The PathSim is an SS measure that captures to what degree two objects (e.g., a pair of tokens in reports) are similar, based on the number of connections between them under a user-defined meta-path in a defined heterogeneous information network. A heterogeneous information network (i.e., G) is a logical network that defines multiple-typed objects and typed relations between typed objects (Sun and Han 2012). The meta-path is a composite relation in a G defined by a user that indicates how two objects can be connected. In this research, the PathSim is selected because: (1) it allows to jointly measure corpus-based and knowledge-based semantic similarities; (2) it is based on user-defined meta-paths that can define how intra-class concepts should be interpreted; (3) it supports the representation of semantic similarities defined by contexts to capture the meaning of text in bridge inspection reports; and (4) it is not biased to information units with high occurrence rates, which allows for capturing semantically-similar "peers" (Sun et al. 2011). Given G and a meta-path (i.e., P), the sematic similarity between two tokens, w_1 and w_2, can be computed as Eq. (4.6) (Sun et al. 2011):

$$S(w_1, w_2) = \frac{2 \times \left|\left\{P_{w_1 \rightsquigarrow w_2} : P_{w_1 \rightsquigarrow w_2} \in P\right\}\right|}{\left|\left\{P_{w_1 \rightsquigarrow w_1} : P_{w_1 \rightsquigarrow w_1} \in P\right\}\right| + \left|\left\{P_{w_2 \rightsquigarrow w_2} : P_{w_2 \rightsquigarrow w_2} \in P\right\}\right|} \tag{4.6}$$

In Eq. (4.6), $P_{w_1 \rightsquigarrow w_2}$ is a path instance between w_1 and w_2 under the defined meta-path P. Given a meta-path P, the number of path instances between w_1 and w_2 is normalized by the total number of path instances between w_1 and w_1 and between w_2 and w_2. The normalization is intended to avoid biases towards tokens with high occurrence rates and to find semantic "peer" tokens.

This thesis proposes to represent the typed objects as a set of preceding tokens, current tokens, succeeding tokens, stems, POS tags, and highest-level concepts in the BridgeOnto. Also, the thesis proposes two basic types of meta-paths, which include: (1) $token \rightarrow stem \rightarrow POS$, and (2) $token \rightarrow stem \rightarrow Onto$. In these two meta-paths, POS denotes a POS tag, $Onto$ denotes a highest-level concept in the BridgeOnto, and $token$ could be a current token or the preceding token or the succeeding token of the current token. Therefore, in the proposed IE model, there are six meta-paths in total to collectively determine token-to-token semantic similarities. Examples of the constructed heterogeneous information network and the defined meta-paths are shown in Figure 4.3.

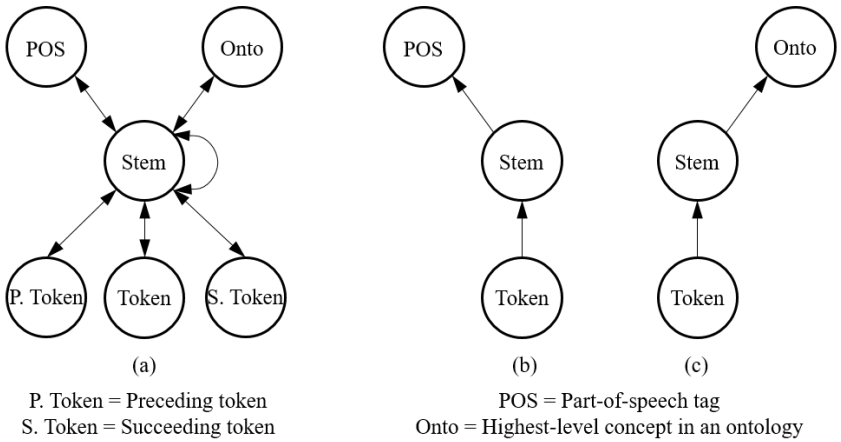


Figure 4.3. Constructed heterogeneous information network (a), and defined meta-paths (b) and (c).

The thesis proposes to linearly combine the six meta-paths, as per Eq. (4.7), to compute a final SS measure for each token pair in the dataset. In Eq. (4.7), PT, CT, and ST denote meta-paths for preceding, current, and succeeding tokens, respectively. For each meta-path P in $\{PT, CT, ST\}$, both POS-based and $Onto$-based meta-paths are considered to compute the SS between w_1 and w_2 . For example, $S_{(PT,POS)}(w_1, w_2)$ denotes the SS between w_1 and w_2 under the $preceding\ token \rightarrow stem \rightarrow POS$ meta-path. In Eq. (4.7), μ controls how much to value the POS-based or $Onto$-based meta-path and ν_P controls how much to value the meta-paths based on token positions (i.e., preceding, current, and succeeding), where $\nu_{PT} + \nu_{CT} + \nu_{ST} = 1$. The best values for μ and ν_P were empirically studied. The experimental results are presented in Sections 4.3.2.1 and 4.3.2.2.

$$S(w_1, w_2) = \sum_{P \in \{PT,CT,ST\}} \nu_P \{\mu \times S_{(P,POS)}(w_1, w_2) + (1 - \mu) \times S_{(P,Onto)}(w_1, w_2)\} \quad (4.7)$$

The proposed IE model derives an entity class sequence for an unlabeled data/sentence by finding the most similar token that was labeled for each token in the unlabeled data/sentence. Then, the entity classes of all the identified and similar tokens that were labeled are sequentially combined to recover an entity class sequence for the unlabeled data. To derive the n^{th} entity class sequences, the n^{th} similar tokens that were labeled and their corresponding entity classes are then used. Although this derivation process assumes independencies between tokens in a sentence, the dependencies in derived entity class sequences are captured because the proposed SS measure, as per Eq. (4.7), already considers dependencies between adjacent tokens. The effect of the number of similar sequences (i.e., N) on the performance of IE was studied. The experimental results are presented in Section 4.3.2.3. This thesis proposes to normalize the likelihood of a derived entity class sequence for a given unlabeled data/sentence by Eq. (4.8), where T is the length of $\mathbf{x}_i$, N is

the number of most similar label sequence(s), and $S_n(w_t,)$ is the computed semantic similarity degree between the t^{th} token in $\mathbf{x}_i$ and its n^{th} similar token that was labeled.

$$P(\tilde{\mathbf{y}}_{i,n}|\mathbf{x}_i, G) = \prod_{t=1}^{T} S_n(w_t,) / \sum_{n=1}^{N} \prod_{t=1}^{T} S_n(w_t,) \quad (4.8)$$

4.2.2 Implementation of the Proposed Method

The following hypothesis was defined: the proposed IE method can achieve the goal of extracting information about existing conditions and performed maintenance actions from bridge inspection reports with reduced human effort as well as a high precision and recall performance, compared to traditional supervised CRF-based IE (the baseline). In order to test this hypothesis, the proposed IE method was implemented and tested in extracting information from 11 bridge inspection reports collected from different state DOTs. The implementation of the proposed IE method is composed of four primary components: data preparation, semi-supervised CRF modeling, training, and evaluation. Figure 4.4 provides an overview of the implementation. A step-by-step illustration for the application of the proposed IE methodology is presented in Figure 4.5.

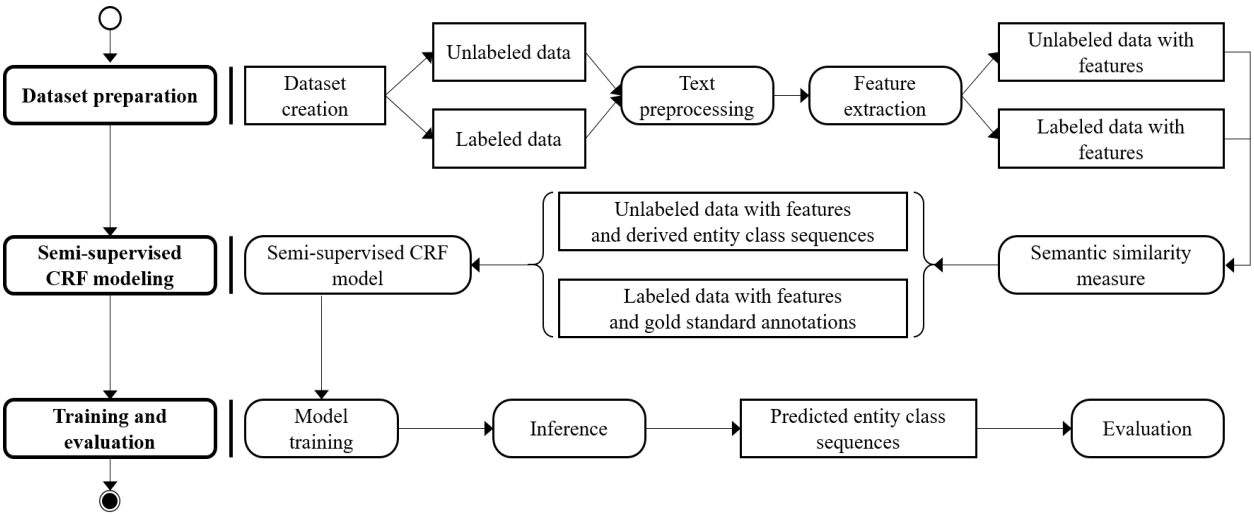


Figure 4.4. Overview of the implementation of the proposed IE method.

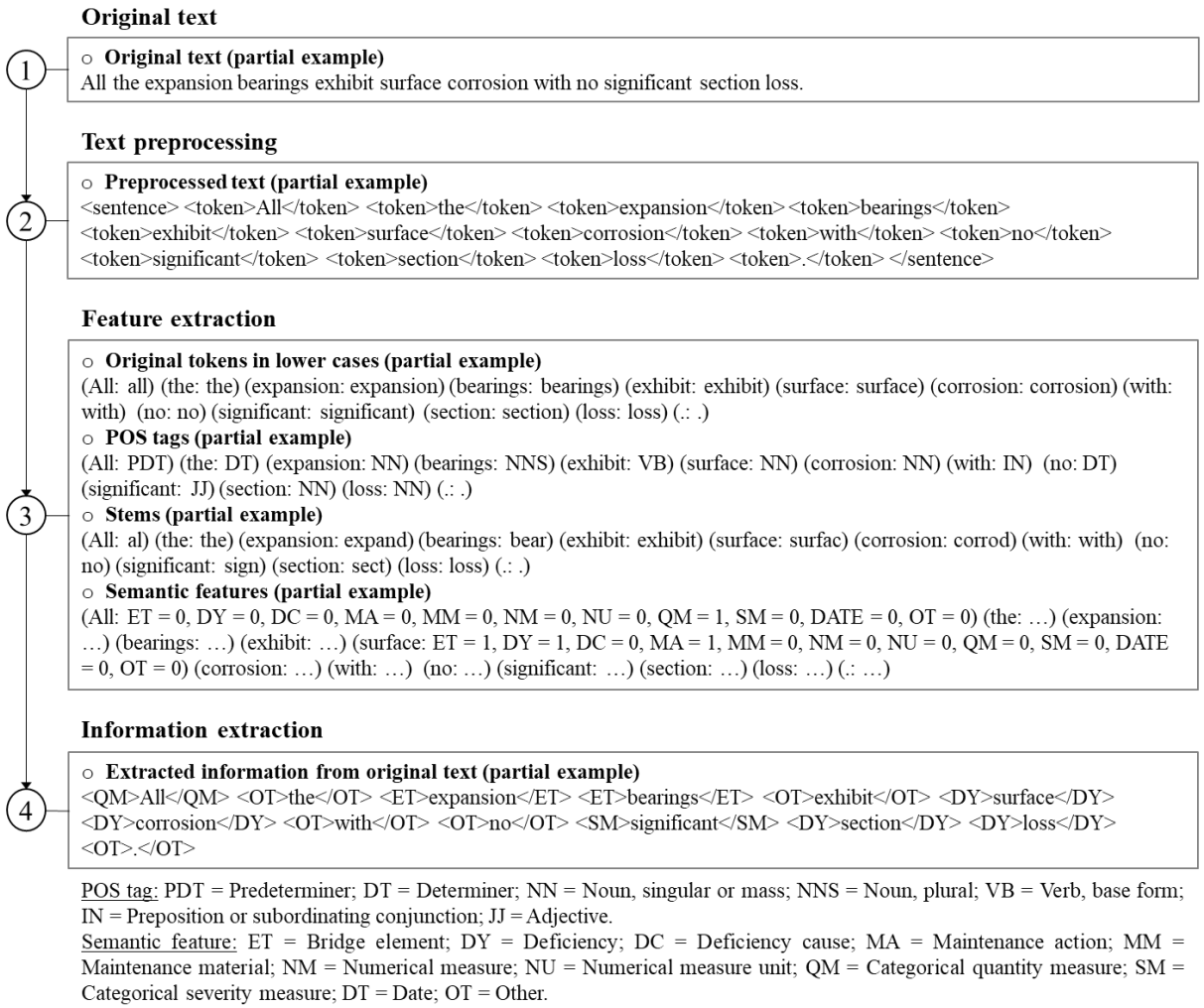


Figure 4.5. An illustrative example for the application of the proposed IE method.

4.2.2.1 Data Preparation

Data preparation aims to (1) create a dataset with annotations for training and evaluation, (2) preprocess the raw text into a format that is ready for further analysis, and (3) extract the features for representing the preprocessed text. Data preparation, thus, includes three subtasks: dataset creation, text preprocessing, and feature extraction.

4.2.2.1.1 *Dataset Creation*

Dataset creation includes dataset selection and human annotation. Dataset selection aims to select two datasets: a labeled dataset (i.e., *L* defined in Section 4.2.1.1) for training and an unlabeled dataset (i.e., *U* defined in Section 4.2.1.1) for extraction and/or evaluation purposes. This thesis proposes to empirically define the best size of labeled data – for the domain text – for achieving an optimal IE performance. The empirical results are presented and discussed in Section 4.3.2.5. Human annotation was conducted by human annotators to mark-up the entire labeled dataset (for training) and the unlabeled dataset (for evaluation) with gold standard labels. The gold standard labels are indicating the true entity class that a token should be classified into. It should be pointed out that the unlabeled data were annotated in this research for evaluation purposes only; in the real-world use of the proposed IE algorithm, users only need to use the small size of fixed labeled/annotated data (i.e., no additional annotations are required). If users want to adapt the proposed algorithm for other types of text and/or for extracting different entities, they would need to develop a small number of fixed annotations based on their different text/entities, and they may annotate another set of unlabeled data for evaluation, if they would like to further test and evaluate the adapted algorithm.

4.2.2.1.2 *Text Preprocessing*

Text preprocessing, also known as text normalization, is a process that converts raw text into a format that is ready for further analysis. Following the standard procedure (Manning and Schütze 1999), the following text preprocessing steps were conducted in this research: tokenization, sentence splitting, and morphological analysis. Tokenization breaks a continuous raw text into a sequence of tokens that include words, digits, punctuations, and whitespaces. Sentence splitting splits the tokenized text into grammar-meaningful sentences by detecting sentence boundaries (i.e.,

sentence-ending characters, such as periods, question marks, etc.). Morphological analysis aims to analyze how a given token is formed based on morphological derivation and inflection and to map the token into its root form. Morphological analysis is needed when mapping a token in text to its corresponding concept in the BridgeOnto for creating semantic features. It was achieved by stemming that removes the suffixes and derivational affixes of a given token to form its stem.

4.2.2.1.3 *Feature Extraction*

Figure 4.2 shows the proposed feature representation for representing each token in a sentence. In this feature representation, the stem features were analyzed and extracted by the Lancaster stemmer on natural language toolkit (NLTK) (Bird et al. 2009). The POS tags were analyzed and extracted by the NLTK POS tagger (Bird et al. 2009). The semantic features were recognized and extracted based on the BridgeOnto (presented in Chapter 3) by comparing the token's stem to the stem(s) of each concept in the ontology.

4.2.2.2 Proposed Semi-Supervised Conditional Random Fields Modeling

The proposed ontology-based, semi-supervised CRF-based IE model is the backbone of the proposed IE method. During the semi-supervised CRF modeling process, with the created datasets, the entity class sequences of the unlabeled data were first derived based on the entity classes of their semantically-similar data that were labeled. The semantic similarities between labeled and unlabeled data were assessed using Eq. (4.7). Then, the normalized likelihood of each derived entity class sequence was computed using Eq. (4.8). Finally, the labeled data in feature representations with their gold standard entity class sequences and the unlabeled data in feature representations with their derived entity class sequences were modeled using Eq. (4.1).

4.2.2.3 Semi-Supervised Conditional Random Fields Training

The training process of the proposed IE method aims to fine-tune the model weight vector $\theta = \{\lambda_k\}_{k=1}^{K}$, so as to maximize the conditional log-likelihood [i.e., $\sum_{i=1}^{l+u} \log P(\mathbf{y}_i|\mathbf{x}_i;\theta)$]. Because of the concavity of the proposed IE model, the training process is guaranteed to reach to a global maximum. In this research, numerical optimization was utilized to fine-tune this weight vector because the conditional log-likelihood cannot be maximized in a closed form (Sutton and McCallum 2006). There are many existing methods for solving this numerical-optimization task, including limited-memory BFGS (Liu and Nocedal 1989), stochastic-gradient descent (Shalev-Shwartz et al. 2011), and average perceptron (Collins 2002), etc.

In this research, the limited-memory BFGS (Liu and Nocedal 1989) was followed for training the proposed IE model. The limited-memory BFGS was selected for four primary reasons: (1) it has been proved to be superior at training L2-regularized log-linear models (i.e., the proposed IE model is one of such models) (Malouf 2002); (2) it is computationally efficient with a limited requirement of computer memory; (3) it does not require heuristically determining the learning rate and the iteration number; and (4) it has high stability (Ngiam et al. 2011).

4.2.2.4 Semi-Supervised Conditional Random Fields Evaluation

The evaluation of the proposed IE method includes two primary components: inference and evaluation. The inference is a process of predicting a structured output (i.e., an entity class sequence) for an unlabeled data/sentence based on the trained model. The evaluation aims to compare the prediction results against the gold standard using evaluation metrics.

4.2.2.4.1 *Inference*

The proposed IE model follows the first-order Markov assumption (i.e., the current entity class depends on the current token and its preceding entity class). This property allows the proposed IE method to use dynamic programming (DP) algorithms to infer structured outputs efficiently. In this thesis, the Viterbi algorithm (Forney 1973) was applied to solve the inference task. This is because the Viterbi algorithm has been widely recognized and successfully applied to many similar supervised ML-based IE tasks. For a given sequence of observations (i.e., tokens in a sentence), each observation could be labeled with any hidden states (i.e., unobserved entity classes). The Viterbi algorithm travels through all the paths constructed by the different combinations of the sequential hidden states to find the most likely entity class sequence for the sentence. For more detailed information on the Viterbi algorithm, the readers are referred to Forney (1973).

4.2.2.4.2 *Evaluation*

To evaluate the performance of the proposed IE method, the precision and recall were selected as the primary evaluation metrics. Precision, as defined in Eq. (4.9) (Olson and Delen 2008), is the percentage of the total number of correctly extracted entities out of the total number of all extracted entities. Recall, as defined in Eq. (4.10) (Olson and Delen 2008), is the percentage of the total number of correctly extracted entities out of the total number of entities that should be extracted. The F-1 measure, as defined in Eq. (4.11) (Olson and Delen 2008), is the weighted harmonic mean of recall and precision. Because the proposed IE method deals with a multi-class classification problem where each token could be labeled with one of the eleven defined entity classes, the average precision, recall, and F-1 measure was also defined as the arithmetic means of precisions, recalls, and F-1 measures over all the entity classes. These evaluation metrics were calculated by comparing the predicted structured outputs with the gold standard annotations. The process for

creating gold standard annotations is presented in Section 4.3.1. The evaluation results are presented and discussed in Section 4.3.2.

$$P = \frac{number\ of\ correctly\ extracted\ entities}{number\ of\ extracted\ entities} \tag{4.9}$$

$$R = \frac{\text{number o}f\ correctly\ extracted\ entities}{number\ of\ entities\ that\ should\ be\ extracted} \tag{4.10}$$

$$F\text{-}1\ measure = 2 \times (P \times R)/(P + R) \tag{4.11}$$

4.3 Information Extraction Method Evaluation

In this research, six primary experiments were conducted to fine-tune the key parameters of the proposed semi-supervised IE algorithm, and to evaluate its performance in supporting automated IE from bridge inspection reports. The first five experiments were conducted to identify the best parameters for: (1) weighting the proposed meta-paths to measure token-to-token sematic similarities, (2) selecting the best number of semantically similar neighbors (i.e., labeled tokens), (3) selecting the best regularization item weight to prevent overfitting or underfitting, and (4) selecting the best size of labeled data (i.e., number of sentences from the selected training text/report) to create the fixed labeled dataset (i.e., L defined in Section 4.2.1.1). The sixth experiment was conducted with the fine-tuned parameters to evaluate the precision and recall of the proposed IE algorithm. These six experiments were conducted following the high-level algorithm shown in Figure 4.1 and the implementation procedure shown in Figure 4.4. The experimental setup, experimental results, and final performance of the proposed IE algorithm are summarized and discussed in the following subsections.

4.3.1 Experimental Setup

In this research, two datasets were developed: a training and development dataset and a testing dataset. The 2006 I-35W Mississippi River Bridge inspection report was selected for creating the training and development dataset. The report was selected, because this bridge experienced a catastrophic collapse in 2007. The report, thus, contains valuable and representative information on various types of deficiencies, maintenance actions, and their related attributes, which would help in training the algorithm and fine-tuning its parameters. Second, 11 other bridge inspection reports from different state DOTs were selected for creating the testing dataset. The 11 reports are considered to be representative because they (1) record bridge conditions at different years by different state DOTs, (2) are for different types of bridges, and (3) exhibit domain-specific complexities with varying text patterns that range from simple to complex ones. A total of 1,866 sentences were randomly collected from these 11 reports. The information about these reports and sentences is presented in Table 4.1.

A manual annotation process was then followed to create the gold standard entity class sequences for training and testing purposes. The goal of the annotation is to assign each token in a sentence to a true entity class by human annotators. The defined entity classes (i.e., target information types) include: bridge element (ET), deficiency (DY), deficiency cause (DC), maintenance action (MA), maintenance material (MM), numerical measure (NM), numerical measure unit (NU), categorical quantity measure (QM), categorical severity measure (SM), date (DT), and other (OT). The collected sentences were separately annotated by five human annotators, who are researchers with background in both civil engineering and NLP. Discrepancies across these five annotation sets were, then, discussed to achieve consensus. Table 4.2 shows example sentences with gold standard annotations.

Table 4.1. Characteristics of the datasets.

Report no.	Reported bridge	State	Year of report	Sentence length*			Number of sentences in report
				Max.	Min.	Avg.	
1	I-35W Bridge	MN	2006	48	3	17	619
2	Natchaug River Chaplin Bridge	CT	2009	68	5	17	112
3	Sherman Minton Bridge	IN	2007	44	5	20	178
4	Hale Boggs Memorial Bridge	LA	2008	55	5	20	255
5	Heron Truss Bridge	MT	2011	72	6	19	287
6	Portsmouth Memorial Bridge	NH	2009	51	6	18	261
7	Wellwood Avenue Bridge	NY	2015	59	4	14	180
8	Union Street Railroad Bridge	OR	2005	64	3	21	178
9	South Park Bridge	WA	2009	42	4	16	138
10	Lower Trento Bridge	NJ	2015	46	7	18	58
11	Raft Island Bridge	WA	2011	42	6	18	100
12	Capitola Crossing Deck Truss	CA	2012	56	3	18	119

* Sentence length is measured by the number of tokens.

Table 4.2. Example sentences with gold standard annotations.

Original sentence	Annotated sentence
The one-half inch thick, oil and stone surface treatment, over two inches of bituminous materials, over a corrugated steel deck, still shows full width transverse cracking, open a maximum of one inch, mainly in the areas of the deck, adjacent to the pier.	The/**OT** one-half/**NM** inch/**NU** thick/**NU** ,/**OT** oil/**MM** and/**OT** stone/**MM** surface/**MA** treatment/**MA** ,/**OT** over/**OT** two/**NM** inches/**NU** of/**OT** bituminous/**ET** materials/**OT** ,/**OT** over/**OT** a/**OT** corrugated/**ET** steel/**ET** deck/**ET** ,/**ET** still/**OT** shows/**OT** full/**SM** width/**SM** transverse/**DY** cracking/**DY** ,/**OT** open/**OT** a/**OT** maximum/**OT** of/**OT** one/**NM** inch/**NU** ,/**OT** mainly/**OT** in/**OT** the/**OT** areas/**OT** of/**OT** the/**OT** deck/**ET** ,/**OT** adjacent/**OT** to/**OT** the/**OT** pier/**ET** ./**OT**
All the expansion bearings exhibit surface corrosion with no significant section loss.	All/**QM** the/**OT** expansion/**ET** bearings/**ET** exhibit/**OT** surface/**DY** corrosion/**DY** with/**OT** no/**OT** significant/**SM** section/**DY** loss/**DY** ./**OT**

4.3.2 Experimental Results and Discussion

4.3.2.1 Knowledge-Based Versus Corpus-Based Semantic Similarity Weight

As mentioned in Section 4.2.1.3, a semantic similarity (SS) measure – including corpus-based and knowledge-based SS indicators – was proposed, as per Eq. (4.7), to measure token-to-token semantic similarities. A set of experiments were conducted to study how different weight combinations [i.e., μ in Eq. (4.7)] of these two types of indicators could affect the IE performance,

so that an optimal weight can be determined and applied for improved performance. A total of 11 controlled experiments were conducted, with the knowledge-based SS weights ranging from 0.0 to 1.0 and with the corpus-based weights ranging from 1.0 to 0.0. The step size for increasing/decreasing the weights is 0.1. The experimental results, in terms of average precision and recall improvements, are shown in Figure 4.6. The average precision/recall improvement is the average precision/recall of the current experiment (i.e., weight combination) minus the minimum average precision/recall among all the controlled experiments.

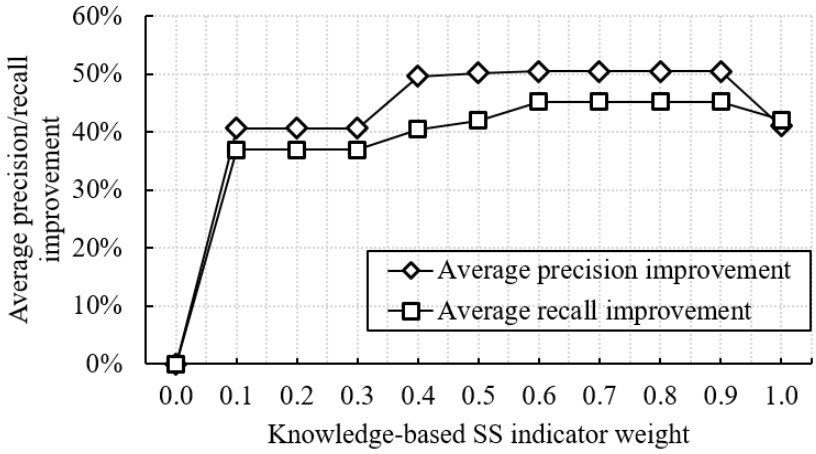


Figure 4.6. Performance of different knowledge-based SS indicator weights.

The experimental results indicate that incorporating domain semantics in semantic similarity assessment can improve the IE performance. As shown in Figure 4.6, increasing the knowledge-based SS weight can increase average precision and recall. For example, the proposed IE algorithm achieved the least satisfactory performance with a 0.0 knowledge-based SS weight, and increasing this weight to 0.9 improved the average precision and recall by 50.5% and 45.2%, respectively. Although increasing the knowledge-based SS weight could improve performance, the optimal IE performance cannot be achieved if semantic features are utilized solely. For example, when the knowledge-based SS was weighted by 1.0, the average precision and recall dropped by 9.4% and

3.1%, respectively, compared to the best performance (shown with a 0.9 knowledge-based SS weight). The corpus-based SS indicator can help when the semantic features cannot differentiate the meaning of a token and thus fail to assess similarity. For example, in the BridgeOnto, the token "concrete" could be treated as a bridge element entity (e.g., "concrete deck") or as a maintenance material entity (e.g., "patched with concrete"). In such situation (although it is relatively rare), the corpus-based SS indicator can be decisive, because the token "concrete" has different POS tags in these two cases (i.e., adjective versus noun).

Figure 4.6 also shows that the proposed IE algorithm is not very sensitive to minor changes in the weights (within 0.1 to 0.9) of both SS indicators. For example, increasing the knowledge-based SS weight from 0.6 to 0.9 did not affect the average precision and recall; however, both averages were improved by around 10% when this weight was increased from 0.1 to 0.9. According to the experimental results and the analysis above, it was concluded that the optimal knowledge-based SS weight is between 0.6 and 0.9 and the optimal corpus-based SS weight is between 0.1 and 0.4 accordingly.

4.3.2.2 Context Window Weight

As mentioned in Section 4.2.1.3, the proposed SS measure, as per Eq. (4.7), also considers the contexts of the current tokens to help with assessing their semantic similarities. The proposed SS measure assumes that the SS indicators between the preceding tokens and between the succeeding tokens are equally important. A set of experiments were conducted to study how to weight the SS indicators [i.e., ν_P in Eq. (4.7)] for the current tokens and for their contexts, in order to collectively determine the SS degree between current tokens for an optimal IE performance. A total of 11 controlled experiments were conducted with the weights for the SS indicator of the current tokens ranging from 0.0 to 0.1 (with a step size of 0.1) and with the weights for the SS indicators of the

contexts ranging from 0.0 to 0.5 (with a step size of 0.05). The experimental results, in terms of average precision and recall improvements, are shown in Figure 4.7.

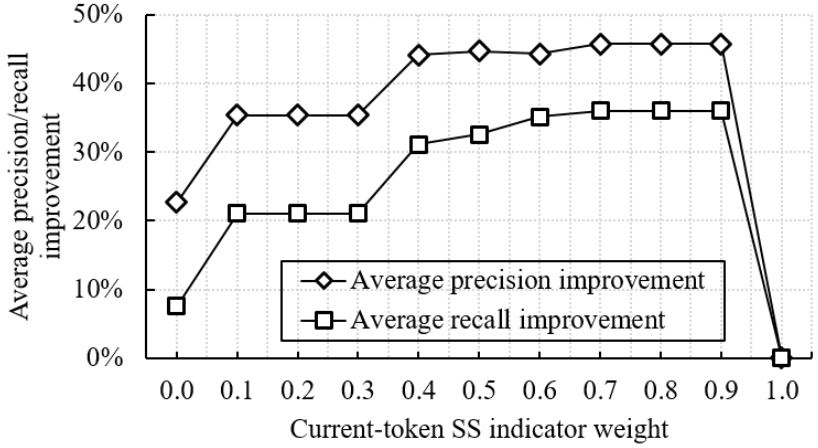


Figure 4.7. Performance of different current-token SS indicator weights.

The experimental results suggest that the semantic similarity of current tokens is important to determine the overall token-to-token SS degree. As seen in Figure 4.7, increasing the current-token SS weight leads to the improvement of average precision and recall. For example, when this weight was set to 0.0, the average precision and recall improved by 22.7% and 7.6%, respectively; but, when this weight was 0.9, the average precision and recall improved by 45.7% and 36.0%. However, overall token-to-token SS could not be purely decided by current tokens. For example, when the current-token SS weight went up to 1.0, the average precision and recall dropped to the least satisfactory performance, which is even worse than the performance without considering current-token semantic similarity. This is because, although the meaning of each current token has been explicitly defined by the BridgeOnto, the contexts of the current tokens can provide useful contextual information that would help when there exist ambiguities in the meanings of current tokens. For example, the token “section” could be considered as a bridge deficiency entity (e.g., “severe section loss”) or as the “other” entity (e.g., “the section of”). Also, they have a same POS

tag of “noun”. In such case, the SS indicators of the contexts could be utilized to assess the semantic similarity of “section” in these two phrases accurately, which would then improve the IE performance.

Figure 4.7 indicates that the proposed IE algorithm is also not very sensitive to minor changes in the weights (within 0.1 to 0.9) of these SS indicators. For example, increasing the current-token SS weight from 0.1 to 0.3 did not cause the average precision and recall to change; but, increasing this weight from 0.1 to 0.9 improved these averages by over 10%. It was concluded that the optimal weight for the SS indicators of the current tokens ranges from 0.7 to 0.9 and the optimal weight for the SS indicators of the contexts ranges from 0.05 to 0.15 accordingly.

4.3.2.3 Number of Similar Neighbor(s)

In the proposed IE algorithm, as per Eq. (4.3), the top N derived entity class sequences of unlabeled data/sentences are used to adapt the algorithm to unseen instances. For each unlabeled sentence, providing more (or less) derived entity class sequences to the proposed IE model could add more (or less) valuable unseen instances, but could also introduce more (or less) noises. To study this tradeoff relationship, 10 controlled experiments were conducted with the number of similar neighbor(s) ranging from 1 to 10 (with a step size of 1). The experimental results, in terms of average precision and average recall improvements, are shown in Figure 4.8.

As indicated by Figure 4.8, with the increase of the number of similar neighbor(s) from 1 to 10, the average precision improvement dropped from 26.4% to 0.0%, and the average recall improvement decreased from 20.2% to 0.0%. Although there exist some outliers (e.g., the average precision improvement at the similar neighbor number of 4), it shows a general trend that increasing the number of similar neighbor(s) decreases the average precision and recall

performance. This observation indicates that the benefits of adding more unseen instances are offset by the noises introduced. Based on the experimental results, it was concluded that the optimal number of similar neighbor(s) for the proposed IE algorithm is 1, which suggests to only use the most similar entity class sequence for each unlabeled data/sentence.

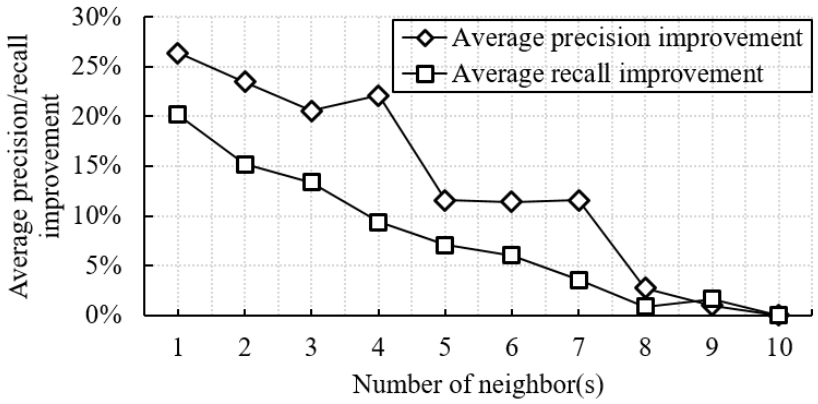


Figure 4.8. Performance of different number of neighbor(s).

4.3.2.4 Regularization Item Weight

In the proposed IE algorithm, the regularization item weight [i.e., γ in Eq. (4.1)] controls how much to penalize each model weight (i.e., λ_k) to prevent overfitting or underfitting. Overfitting could cause a model to be too tailored to noise. Underfitting would make a model fail to capture enough underlying distributions of data. A larger (or smaller) value for γ is likely to cause underfitting (or overfitting). In order to fine-tune this weight, a total of 15 controlled experiments were conducted with the weight ranging from 0.2 to 3.0 (with a step size of 0.2). The experimental results, in terms of average precision and average recall improvements, are shown in Figure 4.9.

The experimental results show that an optimal regularization item weight for the proposed IE algorithm is 0.4. As shown in Figure 4.9, after $\gamma = 0.4$, although the average recall keeps increasing slowly, the average precision starts to drop rapidly. This could be attributed to that a

larger γ underfitted the proposed IE model. When it gets underfitted, the model cannot sufficiently capture relevant relations between input data and target entity classes; it, thus, cannot classify entities precisely (i.e., the number of incorrectly extracted entities increases). In such case, the chances of correctly classifying less-common entities in the dataset (e.g., maintenance action entities) are higher than the chances of correctly classifying commonly-seen entities (e.g., bridge element entities). This causes the increase in average recall of less-common entities to be somewhat larger than the decrease in that of commonly-seen entities. As a result, the average recall could slowly increase as the model gets underfitted and loses precision.

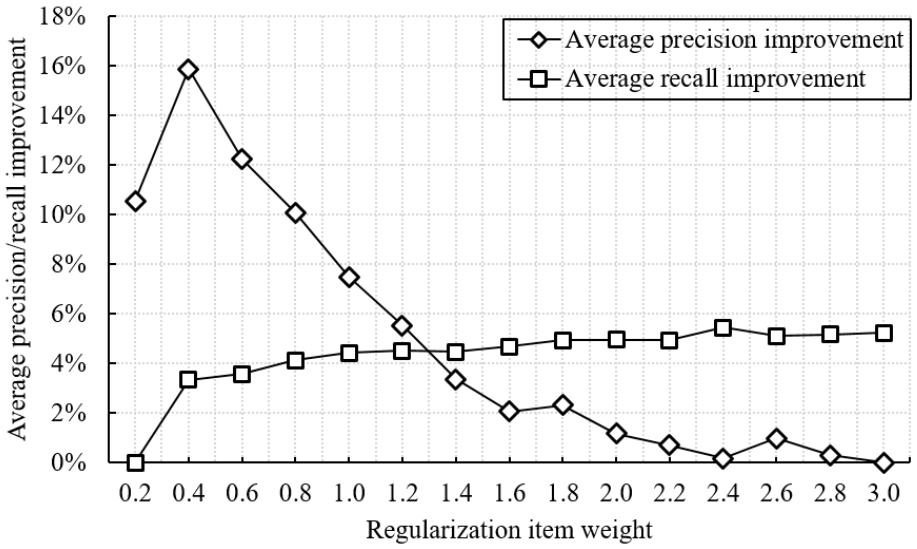


Figure 4.9. Performance of different regularization item weights.

4.3.2.5 Size of Labeled Data

A total of 19 controlled experiments were conducted to study the size of the labeled dataset (i.e., *L* defined in Section 4.2.1.1) and identify the optimal number of the labeled sentences from the I-35W Bridge 2006 inspection report. The first experiment was conducted with 7 sentences; and, the other 18 experiments were conducted with a number of sentences ranging from 35 to 630 (with

a step size of 35). The experimental results, in terms of average precision and recall, are shown in Figures 4.10 and 4.11, respectively.

As seen in Figures 4.10 and 4.11, when 175 sentences were used, the proposed IE algorithm achieved an average precision and recall of 95.5% and 89.3%, respectively. The traditional supervised CRF-based IE (the baseline) only achieved an average precision and recall of 93.7% and 70.1%, respectively, at this number; and, 96.2% and 70.5% respectively with 630 sentences. Although the baseline with 630 sentences is approximately 0.7% higher than the proposed algorithm with 175 sentences in terms of average precision, it is 19.2% lower in terms of average recall. This shows that the proposed IE algorithm outperforms the baseline.

More substantially, as suggested by Figures 4.10 and 4.11, the performances of both algorithms tend to converge or to even drop with the increase in the size of labeled data. For example, average recalls start to drop after 525 sentences, and average precisions show the trend of converging with the increase in size. This could be caused by overfitting to an increasing number of less representative labeled data. This indicates that increasing the size of labeled data does not necessarily improve performance. The key is to increase the comprehensiveness and representativeness of the labeled data, so that an IE algorithm could learn how to deal with different extraction cases. However, as discussed, developing such a labeled dataset is rather challenging and time-consuming. The proposed IE method, by dynamically adapting to the dependency structures and distributions of unlabeled data, offers a promising way to reduce human effort while achieving high performance. Based on the results and analysis above, it was concluded that the optimal number of labeled data/sentences from the I-35W Bridge 2006 inspection report is 175.

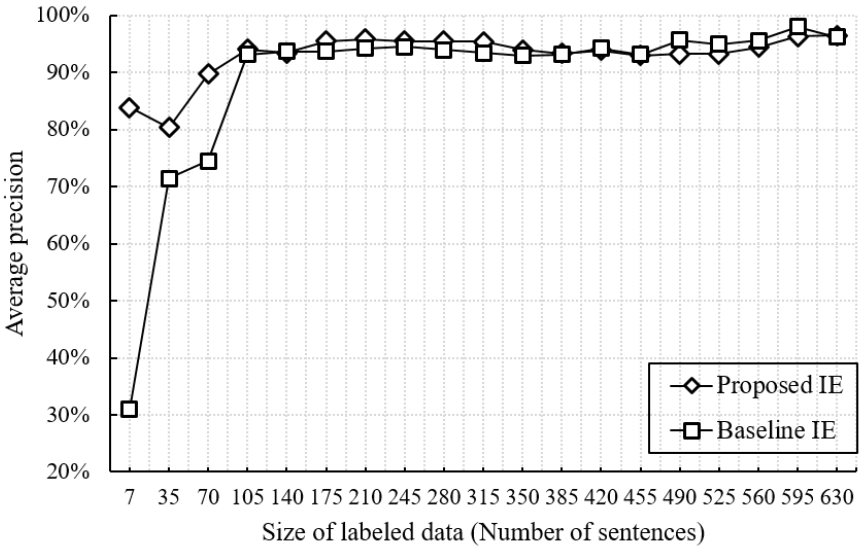


Figure 4.10. Average precision of the proposed and baseline IE algorithms with different sizes of labeled data.

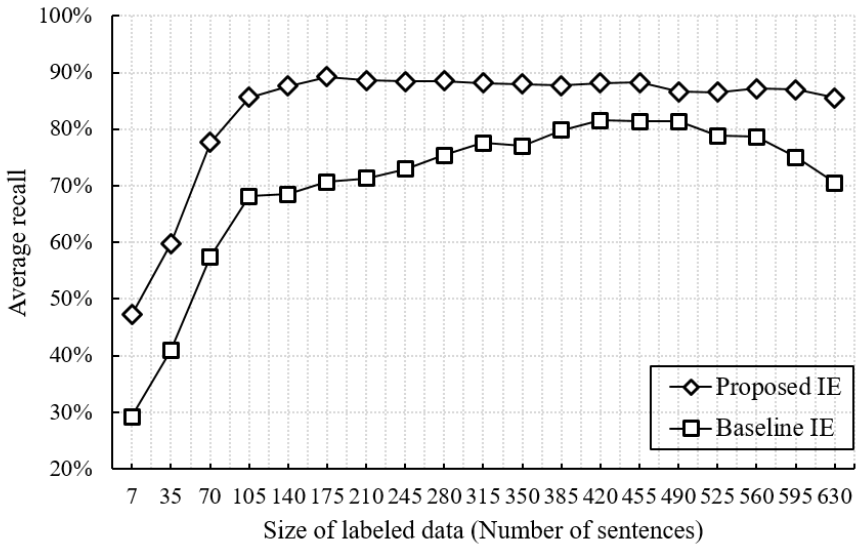


Figure 4.11. Average recall of the proposed and baseline IE algorithms with different sizes of labeled data.

4.3.2.6 Performance of the Proposed Algorithm

This section presents the performance of the proposed IE algorithm. The algorithm was tested on the 11 reports, as per Table 4.1. A total of 175 labeled sentences were used by the proposed algorithm, while a total of 630 labeled sentences were used by the baseline. The parameters that

were used during the testing are shown in Table 4.3. The average precision, recall, and F-1 measure for each testing report are presented in Table 4.4. The processing time for extracting the information was 4 minutes 25 seconds per bridge inspection report, on average, measured on an Intel Core i7 2.2GHz CPU with 16GB RAM.

Table 4.3. Parameters used during evaluation.

Parameter	Parameter value
Knowledge-based semantic similarity weight	0.9
Corpus-based semantic similarity weight	0.1
Current token semantic similarity weight	0.9
Preceding token semantic similarity weight	0.05
Succeeding token semantic similarity weight	0.05
Number of similar neighbor	1
Regularization item weight	0.4
Number of labeled sentences (baseline IE)	630
Number of labeled sentences (proposed IE)	175

Table 4.4 Evaluation results*.

Report	Proposed IE			Baseline IE		
	P	R	F-1	P	R	F-1
1	91.3%	81.6%	86.2%	84.8%	83.4%	84.1%
2	97.7%	78.2%	86.9%	89.3%	72.7%	80.1%
3	96.7%	91.6%	94.1%	89.9%	86.7%	88.2%
4	92.3%	85.3%	88.7%	88.8%	80.6%	84.5%
5	86.0%	77.3%	81.4%	85.5%	80.7%	83.1%
6	91.1%	87.4%	89.2%	89.0%	82.1%	85.4%
7	96.8%	96.3%	96.6%	73.2%	85.4%	78.8%
8	96.5%	88.8%	92.5%	85.8%	80.4%	83.0%
9	95.3%	93.9%	94.6%	91.2%	81.0%	85.8%
10	98.8%	96.9%	97.8%	85.5%	79.1%	82.2%
11	92.8%	87.0%	89.8%	80.8%	76.4%	78.5%
Mean	94.1%	87.7%	90.7%	85.8%	80.8%	83.1%
Standard deviation	0.038	0.068	0.049	0.051	0.039	0.030
Coefficient of variance	0.040	0.077	0.054	0.060	0.048	0.036

*P, R, and F-1 stands for average precision, recall, and F-1 measure, respectively.

As shown in Table 4.4, the proposed algorithm and baseline algorithm achieved an average precision, recall, and F-1 measure of 94.1%, 87.7%, and 90.7%, and 85.8%, 80.8%, and 83.1%, respectively. The standard deviation (SD) of the average precision of the proposed algorithm is

lower (SD = 0.038) than that of the baseline (SD = 0.051), which indicates that the average precision performance of the proposed algorithm is more stable across different reports. The SDs of the recall and F-1 measure of the proposed algorithm (SD = 0.068 and SD = 0.049, respectively) are, however, higher than those of the baseline (SD = 0.039 and SD = 0.030, respectively). This could be caused by the random noises in some reports (such as Report 5) that affected the distributions of unlabeled data. Overall, the proposed IE algorithm outperforms the baseline: it achieves higher average precision, recall, and F-1 measure (8.3%, 6.9%, and 7.6% improvements, respectively), with a lower SD for precision. Also, the coefficient of variances of the average precision, recall, and F-1 measure are all under 1.0, which indicates that the performance of the proposed algorithm is stable across different reports.

The confusion matrix in Figure 4.12 shows the number of entities that were extracted from the 11 reports using the proposed algorithm, as well as the number of gold standard annotations, for each entity class. Figure 4.12 shows that the proposed algorithm has a relatively lower precision for the maintenance action (MA) and maintenance material (MM) entity classes. The proposed algorithm mistakenly extracted 53 and 14 "other" entities as MA entities (total extracted MA entities=349) and MM entities (total extracted MM entities=69), respectively. This could be attributed to the data imbalance issue. In the 11 reports, the number of MM and MA entities (306 entities and 56 entities, respectively) are smaller, compared to the number of bridge element (4,341 entities) and deficiency (1,619 entities) entities. The imbalance in entity number causes the algorithm to focus on minimizing classification errors for the entities with a larger number, while insufficiently considering the errors for the entities with a smaller number. In addition, noises could also negatively affect the algorithm, which results in incorrectly extracted entities. For example, the phrase "drilling of possible stress relief holes" contains noises (i.e., "of possible") that break a

semantically-meaningful maintenance action concept (i.e., “drilling stress relief holes”) into two parts. Because the proposed IE algorithm considers the contexts of tokens, the recognition and extraction of the subject tokens (i.e., “stress relief holes”) could be affected by such noises.

	ET	DY	DC	MA	MM	NM	NU	QM	SM	DT	OT	Rec
ET	4250	3	0	2	1	0	0	0	0	0	85	0.98
DY	18	1518	0	6	0	0	0	0	0	0	77	0.94
DC	0	0	109	0	0	0	0	0	0	0	41	0.73
MA	0	1	0	288	6	0	0	0	0	0	11	0.94
MM	0	0	0	0	48	0	0	0	0	0	8	0.86
NM	0	0	0	0	0	581	0	0	0	0	29	0.95
NU	4	0	0	0	0	0	329	0	0	0	26	0.92
QM	0	0	0	0	0	0	0	86	0	0	3	0.97
SM	7	6	3	0	0	0	0	0	122	0	37	0.70
DT	0	0	0	0	0	0	2	0	0	75	49	0.60
OT	42	20	1	53	14	6	0	0	0	8	24709	0.99
Prec	0.98	0.98	0.96	0.83	0.70	0.99	0.99	1.00	1.00	0.90	0.99	

Figure 4.12. Confusion matrix for all extracted and gold standard entities from the 11 reports.

Figure 4.12 also shows that the deficiency cause (DC), MM, and categorical severity measure (SM) entity classes have relatively lower recall performance. For example, 41 out of the 150 DC entities were not recognized (i.e., extracted as “other”). This is mainly caused by out-of-vocabulary tokens and ambiguities. For example, the token “frozen” should be extracted as a DC entity. However, because the BridgeOnto does not define this token in its deficiency cause hierarchy, the token “frozen” was treated as an out-of-vocabulary token without corresponding semantic features and was thus incorrectly extracted as “other”. Also, ambiguities in the meanings of the concepts

in the BridgeOnto challenge the algorithm in successfully recognizing and extracting information. For example, according to the BridgeOnto, the token “extensive” could be considered as a concept in the SM hierarchy or as a part of the concept “extensive UV radiation” in the deficiency cause hierarchy. Entities with conflicting meanings, especially when they have a same POS tag (i.e., “adjective” for both “extensive”), add to the challenge of correct entity recognition.

Overall, the proposed algorithm performs well; on average, it achieved an average precision, recall, and F-1 measure of 94.1%, 87.7%, and 90.7% respectively, with a fixed set of 175 labeled sentences.

CHAPTER 5 – SEMANTIC RELATION EXTRACTION

This chapter presents the proposed relation extraction method for extracting dependency relations from textual bridge inspection reports to represent the extracted information (extracted as per Research Task #3) in a semantically-rich structured way. The method development and evaluation (Research Task #4) are presented in this chapter.

5.1 Comparison to the State of the Art

Representing the extracted information in a semantically-rich structured way is a challenging task. The words in a sentence are isolated, needing to be linked to form meaningful concepts; and the subsequently-linked concepts are semantically-low, needing to be linked to the associated concepts to form a semantically-rich structured representation of the information. There is, thus, a need for dependency parsing methods to extract dependency relations from the reports, in order to represent the information in the reports in a semantically-rich structured way that is ready for data analytics. For example, this sentence comes from a bridge inspection report (MnDOT 2006): “overlay has some minor spalls and patched areas around the finger joints, and 3,000 LF of transverse cracks”. Dependency parsing is needed to extract the dependency relations to link the words “patched” and “areas” into the deficiency concept “patched_areas”; and then link “patched_areas” to the bridge element concept “overlay”, the categorical severity measure concept “minor”, and the categorical quantity measure concept “some” to represent the sentence in a semantically-rich structured way: <overlay, patched_areas, minor, some>. Without dependency relations, it would be very challenging (if not impossible) to automatically infer from this unstructured sentence which bridge element (i.e., “overlay” or “finger_joints”) has which deficiency (i.e., “spall”, “patched_areas”, or “transverse_crack”) that is “minor” and “some”.

Existing dependency parsing methods are, however, not able to effectively extract dependency relations in such highly technical, domain-specific text – such as that in bridge inspection reports – for two main reasons. First, the current state-of-the-art dependency parsing methods (e.g., Chen and Manning 2014; Dyer et al. 2015; Weiss et al. 2015; Alberti et al. 2015; Zhou et al. 2015; Yazdani and Henderson 2015; Cheng et al. 2016; Kiperwasser and Goldberg 2016; Kuncoro et al. 2017; Hashinmoto et al. 2017; Dozat and Manning 2017; Nguyen et al. 2017; Strubell and McCallum 2017; Babbar and Schölkopf 2017) mostly rely on a single machine learning classifier to extract dependency relations. A single classifier is not sufficient in capturing the complex configuration distributions of the text in the bridge reports, because the reports are written by many different writers/inspectors from various agencies and are thus highly-variable in terms of text characteristics and patterns. An ensemble of classifiers usually performs better than a single classifier (Babbar, R., and Schölkopf 2017; Zhang et al. 2011; Schiele 2002; Dietterich 2000), especially when dealing with highly-dimensional data (Pes et al. 2017; Yu et al. 2017) and/or data with complex distributions such as imbalanced distributions (Haixiang et al. 2017; Bickel et al. 2007; Sun et al. 2006). Second, existing dependency parsing methods (e.g., Chen and Manning 2014; Dyer et al. 2015; Weiss et al. 2015; Alberti et al. 2015; Zhou et al. 2015; Yazdani and Henderson 2015; Cheng et al. 2016; Kiperwasser and Goldberg 2016; Kuncoro et al. 2017; Hashinmoto et al. 2017; Dozat and Manning 2017; Nguyen et al. 2017; Strubell and McCallum 2017; Babbar and Schölkopf 2017) typically only use syntactic features for supporting the extraction of dependency relations. But, semantic text features are also very important for facilitating dependency parsing, because they provide semantics on word-to-word interactions that are critical when deciding on how sentences should be parsed. For example, based on the defined semantics that a categorical severity measure describes a bridge deficiency, the dependency

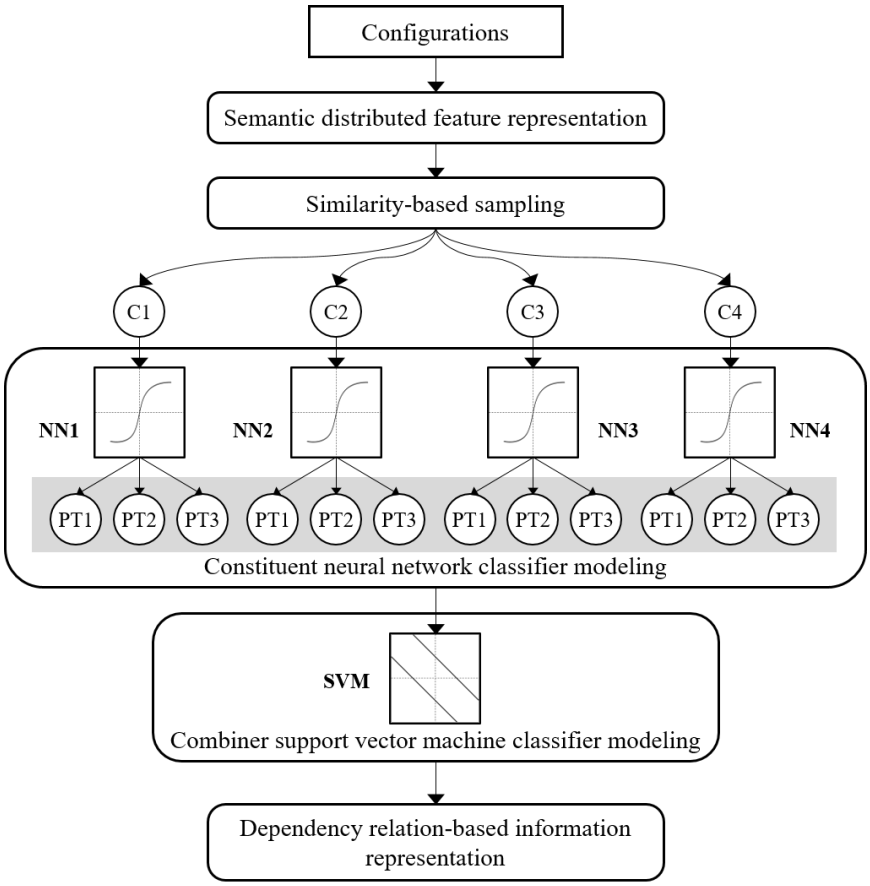
relation between the concepts “minor” (as a modifier) and “patched_areas” (as a head) can be analyzed and extracted correctly.

5.2 Relation Extraction Method Development

5.2.1 Proposed Relation Extraction Method

To address the aforementioned knowledge gaps, a semantic neural network ensemble (NNE)-based relation extraction [i.e., dependency parsing (DP)] method is proposed. The proposed method is composed of five primary components, as per Figure 5.1: semantic distributed feature representation, similarity-based sampling, constituent NN classifier modeling, combiner SVM classifier modeling, and dependency relation-based information representation. The proposed method is novel in three primary ways. First, it proposes a new feature representation for the configurations, which includes both syntactic (words and POS tags) and semantic (the semantic classes of words) text features. The semantic features aim to capture the semantics about the word-to-word interactions for facilitating the extraction of dependency relations. Second, it proposes and utilizes a new similarity-based sampling method to capture the distribution characteristics of the configurations and sample the similarly-distributed configurations into the same clusters. Compared to existing sampling methods used in ensemble learning (see Section 2.3.3), the proposed method can better capture how the configurations distribute. It generates more meaningful configuration clusters that contain the densely- and sparsely-distributed as well as the correctly and incorrectly densely-distributed configurations, which facilitates the classifier ensembling and the NNE-based DP. Third, the proposed DP method takes an ensemble learning-based approach. It uses a set of constituent NN classifiers to collectively capture the complex distributions of all the configurations, and utilizes a combiner SVM classifier to capture the classification and/or misclassification patterns of the NN classifiers for making final predictions

on the transition types. Each of the constituent classifiers only learns from similarly-distributed and thus more easily-separable configurations. The ensemble of the classifiers can better capture the complex distributions, which are challenging for a single classifier to capture (Haixiang et al. 2017; Bickel et al. 2007; Sun et al. 2006).



C1 = majority cluster; C2 = minority cluster; C3 = correct-majority cluster; C4 = incorrect-majority cluster; NN = neural network classifier; PT = probability of a transition type ("Shift", "Right-arc", or "Left-arc"); SVM = support vector machines classifier.

Figure 5.1. Proposed semantic neural network ensemble (NNE)-based dependency parsing (DP) method.

5.2.1.1 Semantic Distributed Feature Representation

A new semantic distributed feature representation is proposed to represent the configurations. As shown in Figure 5.2, it is a multi-level representation. First, the configurations are represented by

the configuration-based features. These features are defined according to the positions of the elements (words of a sentence) in a configuration (Zhang and Nivre 2011). The configuration-based features include 14 features: (1) the top three elements of the stack: σ_1, σ_2, σ_3; (2) the top three elements of the buffer: β_1, β_2, β_3; (3) the first and second leftmost/rightmost children of the first element in the stack: $lc_1(\sigma_1)$, $rc_1(\sigma_1)$, $lc_2(\sigma_1)$, $rc_2(\sigma_1)$; and (4) the first and second leftmost/rightmost children of the second element in the stack: $lc_1(\sigma_2)$, $rc_1(\sigma_2)$, $lc_2(\sigma_2)$, $rc_2(\sigma_2)$.

Second, each of the configuration-based features is represented by syntactic and semantic text features. The syntactic features include: (1) words: the original lexical forms of the words; and (2) POS tags: the lexical classes of the words, which are defined based on the syntactic structures of the sentences. The semantic features are the semantic classes of the words. In this research, to capture the semantics about the word-to-word interactions in the text and the information that needs to be extracted and represented, the following semantic classes were defined based on an analysis of a sample of bridge inspection reports: bridge element (ET), deficiency (DY), deficiency cause (DC), numerical measure (NM), numerical measure unit (NU), categorical quantity measure (QM), categorical severity measure (SM), maintenance action (MA), maintenance material (MM), and date (DT).

Third, the text features are further represented using distributed feature representations. For example, instead of using "noun" as the POS tag for the word "crack", the NN-based distributed feature representation utilizes a vector with a user-defined vector size to represent it numerically. Thus, using the proposed feature representation, a configuration is represented by a numeric vector of size 2100: 14 configuration-based features, 3 text features for each of the configuration-based features, and a vector of size 50 for each of the text features.

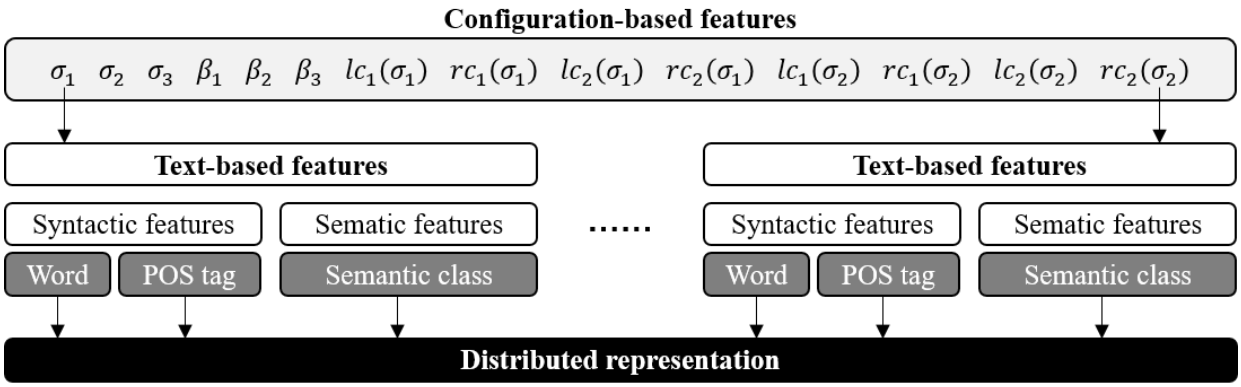


Figure 5.2. Proposed semantic distributed feature representation.

5.2.1.2 Similarity-Based Sampling

The configuration distributions of the text exhibit the following characteristics: (1) a majority of the configurations are distributed in a dense area; (2) a minority of them are distributed in a sparse area; and (3) in the dense area, the configurations of a gold standard transition (GST) type ("shift", "left-arc", or "right-arc") overlap with the configurations of the other GST types. Because of the overlapping, some of the configurations distribute relatively far away from the center of their corresponding GST type and relatively close to one of the other centers, where a center is the arithmetic mean of all the configurations that belong to the same GST type. To capture these characteristics in a way that each constituent NN classifier will be trained only with the similarly-distributed and thus more easily-separable configurations, this thesis proposes to sample the configurations into one or two of the following four configuration clusters:

- *C1*: This is a majority cluster, which contains the densely-distributed configurations that belong to all the GST types and are distributed close to one of the centers, where $C1 = C3 \cup C4$.
- *C2*: This a minority cluster, which contains the sparsely-distributed configurations that belong to all the GST types and are distributed far away from all the centers.

- *C3*: This a correct-majority cluster, which contains the densely-distributed configurations that belong to all the GST types and are distributed close to the center of their corresponding *correct* GST type, where $C3 \in C1$.
- *C4*: This an incorrect-majority cluster, which contains the densely-distributed configurations that belong to all the GST types and are distributed close to the center of another *incorrect* GST type, where $C4 \in C1$.

The C1 and C2 clusters aim to differentiate the densely-distributed configurations from the sparsely-distributed ones. The C3 and C4 clusters aim to differentiate the densely-distributed configurations in C1 – differentiating those that distribute close to the center of their correct GST type from those that distribute close to the center of an incorrect GST type.

To sample all the configurations into the aforementioned clusters, a similarity-based sampling method is proposed. Similarities between the configurations and their centers are indicative of the distribution characteristics. For example, if a configuration is similar to (is close to) the center of a GST type, it is sampled into the C1 cluster. However, similarity measured in one feature space is insufficient to capture the complex distribution characteristics, because different degrees of similarities (measured distances for indicating "being close" or "being far away") emerge when some other features are used for the measurement (Harispe et al. 2015). To deal with this issue, this thesis proposes to measure the similarities in seven different feature spaces, and utilize the similarities measured in these spaces collectively as a criterion to sample the configurations into the defined clusters. These feature spaces are defined in Table 5.1.

Table 5.1. The defined feature spaces.

Feature space	Features [a]
1	Words
2	POS tags
3	Semantic classes
4	Words + POS tags
5	Words + semantic classes
6	POS tags + semantic classes
7	Words + POS tags + semantic classes

[a] All the features are in their distributed representations; POS = part-of-speech.

The proposed similarity-based sampling method is summarized as follows. First, the centers of the configurations of the three GST types are computed in each feature space. A center is computed by calculating the arithmetic mean of all the configurations (in the proposed feature representation) that belong to the same GST type and are in the same feature space. As a result, a total of 21 centers (for 3 GST types and 7 feature spaces) are generated. Second, a configuration gets associated with a similarity-based transition (ST) type in each feature space. In a space, the ST type of a configuration is the GST type of the center that is most similar to the configuration compared to the other two centers, where the similarity degree is computed by the cosine-similarity measure. The ST and GST types of a configuration could be same or different, because the configuration could be closer to the center of a correct or an incorrect GST type. As a result, a configuration gets associated with a total of 7 ST types, one per feature space. Third, the configurations are sampled into the above-defined clusters based on Eq. (5.1), where $CMST$ is the count of the majority ST, MST is the type of the majority ST, and GST is the gold standard transition type.

For a configuration, Eq. (5.1) works as follows. First, if the CMST of the configuration is greater than or equal to the 'natural threshold' (i.e., 4 out of 7), the configuration is sampled into the *C1* cluster; otherwise, it is sampled into the *C2* cluster. This is because in the former case the majority of the STs have reached a consensus, which indicates that the configuration can be confidently associated close to one of the centers; while in the latter case no consensus has been made, which

indicates that the configuration cannot be confidently associated close to any of the centers. Second, if a *C1* configuration happens to have an MST type that is same as its GST type, it is sampled into the *C3* cluster as well; otherwise, it is sampled into the *C4* cluster. This is because in the former case the majority of the STs are indicating a correct transition type (the GST type of the configuration), while in the latter case no correct transition type can be decided based on the MST type.

$$\text{Cluster} = \begin{cases} \text{C1} & \text{if CMST} \geq 4; \\ \text{C2} & \text{if CMST} < 4; \\ \text{C3} & \text{if CMST} \geq 4 \text{ and MST} = \text{GST}; \\ \text{C4} & \text{if CMST} \geq 4 \text{ and MST} \neq \text{GST}. \end{cases} \quad (5.1)$$

5.2.1.3 Constituent Neural Network Classifier Modeling

An NN architecture was modeled and developed for training a set of constituent NN classifiers. It is a feedforward neural network that contains an input layer, a hidden layer, and an output layer. This NN architecture was chosen for two reasons. First, it can automatically learn the most-useful feature conjunctions and high-order features, which helps avoid feature sparsity and incompleteness issues (Chen and Manning 2014; Mikolov et al. 2013). Second, it does not use a complex neural network topology, which helps balance classification accuracy and computational efficiency (Chen and Manning 2014).

The input layer takes the semantic distributed feature representation of a configuration as input. A unit of the input layer takes a value from the representation. Based on the size of the semantic feature representation vectors (see Section 5.2.1.1), the input layer has a size of 2100. The hidden layer contains a set of hidden units, each of which is fully connected to the input layer. A hidden unit takes a value mapped from the input layer. The mapping is conducted by an activation function. For instance, using the logistic sigmoid function as an example, a hidden unit has an

input value of h_i that is computed by Eq. (5.2) (Zadeh et al. 2010), where $W_1 \in R^{|X|\times|H|}$ is a weight matrix, $B_1 \in R^{|H|}$ is a bias vector, $|X|$ is the size of the input layer, and $|H|$ is the size of the hidden layer. In this research, a hidden layer size of 200 (Chen and Manning 2014) was used, and the logistic sigmoid function was selected and used based on the experimental results (see Section 5.3.1). The output layer is a softmax layer added upon the hidden layer and is used to model the multi-class probabilities of a configuration being classified into the transition types. The probabilities are computed by Eq. (5.3) (Bouchard 2007), where t_j is the j^{th} transition type, $W_2 \in R^{|3|\times|H|}$ is a weight matrix, and $B_2 \in R^{|3|}$ is a bias vector. Based on the number of transition types in the transition-based DP model, the output layer has a size of 3. For a dataset $D = \{(c^k, t^k)\}_{k=1}^{K}$, where c^k is the k^{th} configuration and t^k is its corresponding GST type, the training process of the NN architecture aims to minimize the L_2 -regularized cross-entropy loss (maximizing the probabilities of the training configurations being classified into their GST types). The loss function is defined in Eq. (5.4) (De Boer et al. 2005), where $\theta = \{W_1, B_1, W_2, B_2\}$ and λ is a regularization parameter.

$$h_i = \frac{1}{1+\exp(-W_{1i}X - B_{1i})}, i = 1, \dots, |H| \tag{5.2}$$

$$P_{t_j} = \frac{\exp(W_{2j}h + B_{2j})}{\sum_{j=1}^{j=3}\exp(W_{2j}h + B_{2j})}, j = 1, 2, 3 \tag{5.3}$$

$$L(\theta) = -\sum_{k} \log P_{t_k} + \frac{\lambda}{2}\|\theta\|^2 \tag{5.4}$$

5.2.1.4 Combiner Support Vector Machine Classifier Modeling

A combiner SVM classifier was modeled and developed. It aims to capture the misclassification and/or classification patterns of all the constituent NN classifiers, and to make final configuration

classification decisions for extracting dependency relations from the text. As shown in Figure 5.1, the combiner SVM classifier takes the outputs of the four constituent NN classifiers (three probabilities per constituent classifier; see Section 5.2.1.3) as input. Thus, the input of the combiner classifier is a probability vector of size 12. Training a classifier in such case is a straightforward learning process, because the input contains less features and simple patterns and the resulting learning process does not involve extensive feature conjunctions and mappings. SVM has shown high performance in such learning tasks (e.g., Priya and Aruna 2012; Shibuya et al. 2015)), and was therefore chosen for training the combiner classifier.

5.2.1.5 Dependency Relation-Based Information Representation

A dependency relation-based information representation method is proposed. It aims to decode the extracted word-to-word dependency relations, in order to link the isolated words into semantic information elements (SIEs) and to represent the unstructured and semantically-low SIEs into semantically-rich structured semantic information sets (SISs). In this research, an SIE is a concept that describes bridge conditions and maintenance actions, which could be a bridge element (ET), deficiency (DY), deficiency cause (DC), numerical measure (NM), numerical measure unit (NU), categorical quantity measure (QM), categorical severity measure (SM), maintenance action (MA), maintenance material (MM), or date (DT). An SIS is a semantic information structure that consists of SIEs. The SIEs in an SIS must follow an SIE-to-SIE dependency relation type. For example, as illustrated in Figure 5.3, the SIEs must follow one of the three SIE-to-SIE dependency relation types: (1) the ET-DY dependency relation with the semantics: a "*bridge element"* is affected by a "*deficiency"* that is inspected at a *"date"*, that is caused by a "*deficiency cause"* and is maintained by a "*maintenance action"* using a "*maintenance material"*, and that has a "*numerical measure"* with a "*numerical measure unit"*, a "*categorical severity measure"*, and a "*categorical quantity*

measure"; (2) the ET-DC dependency relation with the semantics: a "*bridge element"* has a "*deficiency cause"* that is inspected at a *"date"*; and (3) the ET-MA dependency relation with the semantics: a "*bridge element"* is maintained by a *"maintenance action"* using a *"maintenance material"* at a *"date"*.

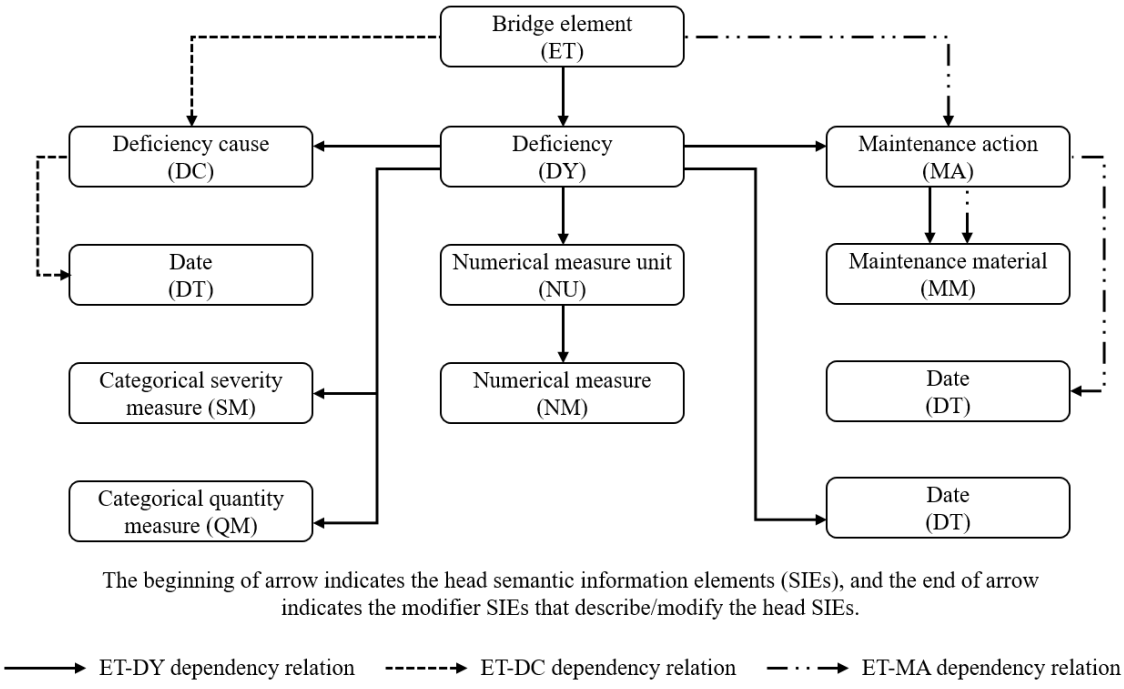


Figure 5.3. SIE-to-SIE dependency relations defined in semantic information set (SIS).

The proposed dependency relation-based information representation method, as illustrated in Figure 5.4, contains three main steps. First, a sentence is represented with a sequence of words, semantic classes, word numbers, and head word numbers. The word and head word numbers indicate the extracted word-to-word dependency relations. For example, in Figure 5.4, the head word of "chord" is at position 4, which is "connection". Second, the modifier and the corresponding head words, as well as their semantic classes, are combined to form SIEs and the semantic classes of the SIEs, respectively. In this step, only the word and the head word numbers of the original head words are maintained. For example, in Figure 5.4, the modifier words "bottom"

and “chord” were combined with the head word “connection” to form the SIE (i.e., “bottom chord connection”) and the semantic class (i.e., “ET”) with the word and head word numbers of 4 and 7, respectively. The semantic classes of the SIEs are needed in order to associate the right SIEs into the right positions of an SIS, and to break down the SIEs that contain concepts with different semantic classes. For example, in Figure 5.4, the phrase “severe crevice corrosion” was further broken down into “severe” and “crevice corrosion” SIEs based on their semantic classes (“SM” and “DY”, respectively). Third, the extracted SIE-to-SIE dependency relations are checked to assess whether they follow the SIE-to-SIE dependency relations as defined in Figure 5.4, so that only valid SIEs are added to an SIS. For example, in Figure 5.4, the SIE pair “bottom chord connection” and “truss” was excluded because there are no dependency relations defined between the two ET SIEs.

Step 3	**SIS**	bottom_chord_connection ET							severe SM	crevice_corrosion DY	
Step 2	**HN**	7					4	0	10	7	
	SIE	bottom_chord_connection					truss	has	severe	crevice_corrosion	
	SC	ET					ET	OT	SM	DY	
	WN	4					6	7	8	10	
Step 1	**HN**	4	4	4	7	6	4	0	10	10	7
	WD	the	bottom	chord	connection	of	truss	has	severe	crevice	corrosion
	SC	OT	ET	ET	ET	OT	ET	OT	SM	DY	DY
	WN	1	2	3	4	5	6	7	8	9	10

SIS = Semantic information set; SIE = Semantic information element; HN = Head word number; WD = Word; SC = Semantic class; WN = Word number; OT = Other; ET = Bridge element; SM = Categorical severity measure; DY = Deficiency.

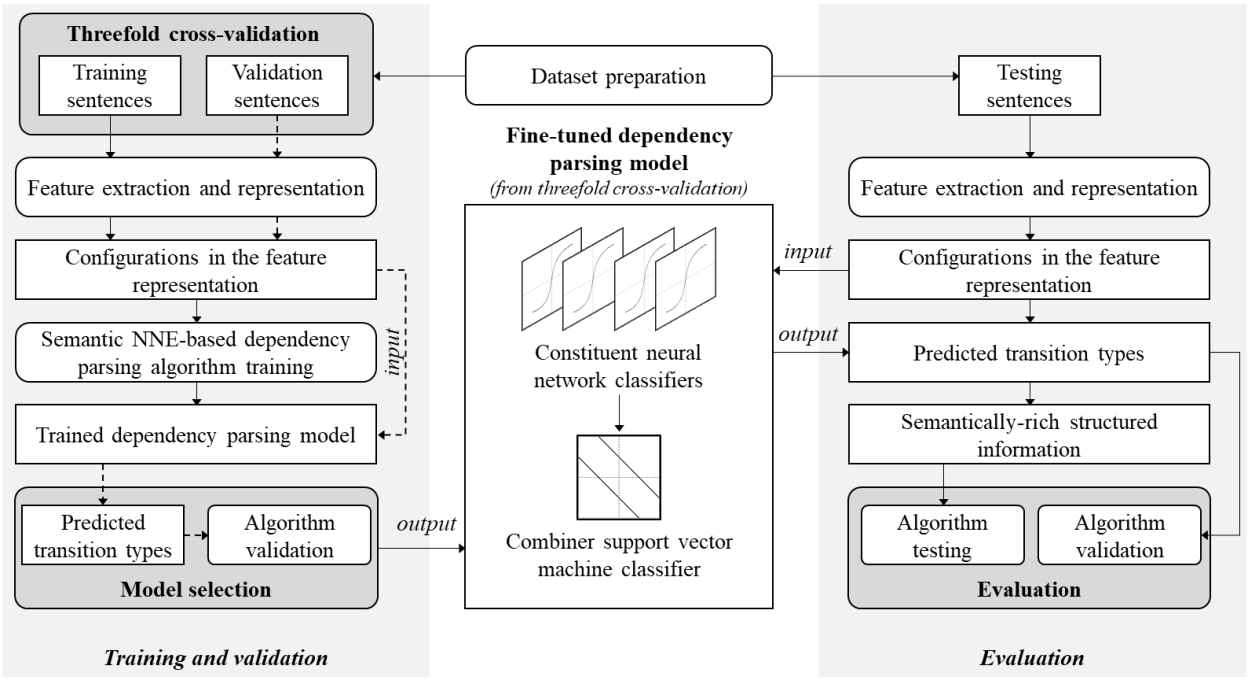
Figure 5.4. An example to illustrate the proposed dependecy relation-based information representation method.

Two special cases that include conjunction and negation are also considered in the proposed information representation method. First, if one SIE is dependent on the other SIE and they are concatenated by a conjunction, they inherit the dependency relations of each other. For example, in the following sentence from (WSDOT 2009), “abutment” and “deck” have a dependency

relation and are concatenated by a conjunction (i.e., “and”): “Leaching at corner of north abutment and bottom of deck.” In this case, “deck” inherits the dependency relations of the “abutment” and gets associated with “leaching” as well. Second, if an SIE is concatenated to another SIE by a negation, both SIEs (and their associated SIEs) are excluded from an SIS. For example, in the following sentence from (Caltrans 2012), “connections” was concatenated with “distress” by a negation (i.e., “did not”): “The connections did not appear to be in distress.” In this case, these SIEs are excluded from an SIS. To capture conjunctions and negations, two gazetteer lists were developed and used. The conjunction gazetteer list includes words/phrases like “and”, “as well as”, “along with”, “together with”, etc. The negation gazetteer list includes words/phrases like “no”, “not”, “doesn’t”, “isn’t”, etc.

5.2.2 Implementation of the Proposed Method

The proposed semantic NNE-based dependency parsing (DP) method was implemented in extracting dependency relations from bridge inspection reports for linking the isolated words that describe bridge conditions and maintenance actions into SIEs and SISs. The implementation included four primary steps: dataset preparation, feature extraction and representation, semantic NNE-based DP algorithm training, and evaluation. An overview of the implementation methodology is presented in Figure 5.5.



NNE = neural network ensemble.

Figure 5.5. Overview of the implementation of the proposed semantic neural network ensemble (NNE)-based dependency parsing (DP) method.

5.2.2.1 Dataset Preparation

Dataset preparation included dataset creation, text preprocessing, and human annotation. A dataset, which contains a total of 1,000 sentences that were randomly selected from 10 bridge inspection reports, was created. As shown in Table 5.2, the selected reports are from different states, from different reporting years, and for different bridge structure types. The sentences were randomly selected to avoid introducing bias. To further ensure that the sample (the selected sentences) is representative of the population (all the sentences from the 10 reports), the distributions of the sentence lengths were compared. As shown in Figure 5.6, the two distributions are quite similar. The p-value for the comparison of the distributions is 0.4892 (calculated from the Welch's unequal variance t-test, assuming normal distributions of the sentence lengths), which shows that there is no significant difference between the two. These results indicate that the sample is representative.

The sentences were further randomly split into three sets at a ratio of 2:1:1 – a training set for algorithm training, a validation set for hyperparameter tuning and algorithm validation, and a testing set for testing the fine-tuned model. As noted above, 50% of the data were used for training, in order to keep the remaining portion of the data for validation and testing. In the initial method development efforts, the use of 75% of the data for training was also tested, which only marginally changed the parsing performance. This indicates that the increase in the ratio of training data, beyond 50%, does not have a substantial impact on the performance results. Table 5.3 shows a set of sentence examples. Text preprocessing aimed to transform the raw text (the selected sentences) into the format required for dependency relation extraction. Tokenization was used to break down a continuous sentence into a sequence of tokens (e.g., words, digits, punctuations, and whitespaces). Human annotation aimed to mark up the entire dataset with gold standard dependency relations. Following the universal dependencies guideline (Marneffe et al. 2014), the sentences were, separately, annotated by three annotators. The three are researchers with background in both civil engineering and natural language processing. The final gold standard annotation was achieved with full agreement of all the annotators.

Table 5.2. List of bridge inspection reports.

No.	Reported bridge	Structure type	State	Year
1	Natchaug River Chaplin Bridge	Concrete arch bridge	CT	2009
2	Sherman Minton Bridge	Double-deck through arch bridge	IN	2007
3	Hale Boggs Memorial Bridge	Cable-stayed bridge	LA	2008
4	Heron Truss Bridge	Steel deck truss bridge	MT	2011
5	Portsmouth Memorial Bridge	Vertical-lift bridge	NH	2009
6	Wellwood Avenue Bridge	Concrete arch bridge	NY	2015
7	Union Street Railroad Bridge	Vertical-lift, Pratt through truss bridge	OR	2005
8	South Park Bridge	Scherzer rolling lift double-leaf bascule bridge	WA	2009
9	Lower Trenton Bridge	Through truss bridge	NJ	2015
10	Capitola Crossing Deck Truss	Single-span deck truss bridge	CA	2012

Table 5.3. Examples of sentences in the bridge inspection reports.

Report no.[a]	Sentence no.	Original sentence from bridge inspection report
1	1	The one-half inch thick, oil and stone surface treatment, over two inches of bituminous materials, over a corrugated steel deck, still shows full width transverse cracking, open a maximum of one inch, mainly in the areas of the deck, adjacent to the pier.
1	2	The outside fascia deck edge plates still show light to moderate rusting, along their edges.
2	3	Several of the anchor bolts for the cross girder bearings on the pier columns exhibit deficiencies that include mis-drilled holes, bent anchor bolts, improperly installed anchor bolts, and loose nuts.
2	4	The curb faces on the westbound deck have minor widespread spalling.
3	5	Throughout the bridge, the bolted field splices for the deck exhibited isolated instances of loose bolts, missing nuts, and missing bolts (see photo 15).
3	6	Rodents, rodent's dens, and moderate rodent debris were noted in tiers 23-25 of both towers.
4	7	The Pier 1 expansion bearing assemblies exhibited approximately 25 percent loss of protective coating with moderate corrosion and negligible loss of section on the exposed areas.
4	8	The timber deck members were coated with creosote and tar.
5	9	Truss bottom chord members typically have deterioration with section loss at the gusset plates and some surface rust throughout webs and top flanges.
5	10	Minor corrosion and section loss of bottom flange angles.
6	11	The underside of the cap beam between columns C1 and C2 exhibits 3' x 20" x 3" deep spall with two main rebars exposed and one stirrup exposed, 32" x 16" x 2" deep spall and hollow sounding concrete areas 12" x 18".
6	12	The joint seal has detached from the joint.
7	13	The paint system of this section appears to be in fair condition overall, with failure on approximately 20 percent of the surface area.
7	14	Some of the rivet heads in these locations have also suffered some moderate section loss.
8	15	South abutment settled downward and retaining walls of abutment rotated outward, allowing span between abutment and bent 2 to settle as well during earthquake.
8	16	Large spall 18" x 24" on west wall of north abutment.
9	17	Several anchor bolts and many keeper plates were noted to be missing at the abutment bearings and a steel bolster Girder 2 exhibits section loss.
9	18	The abutment rocker bearings exhibit pack rust between the masonry plate and the rocker.
10	19	At connections there was generally minor crevice corrosion between the eyebar heads and the pin with an average section loss of approximately 1/8" around the interior circumference.
10	20	Significant section loss to the bottom lacing was found in spots along the top chord.

[a] The report number follows that defined in Table 5.2.

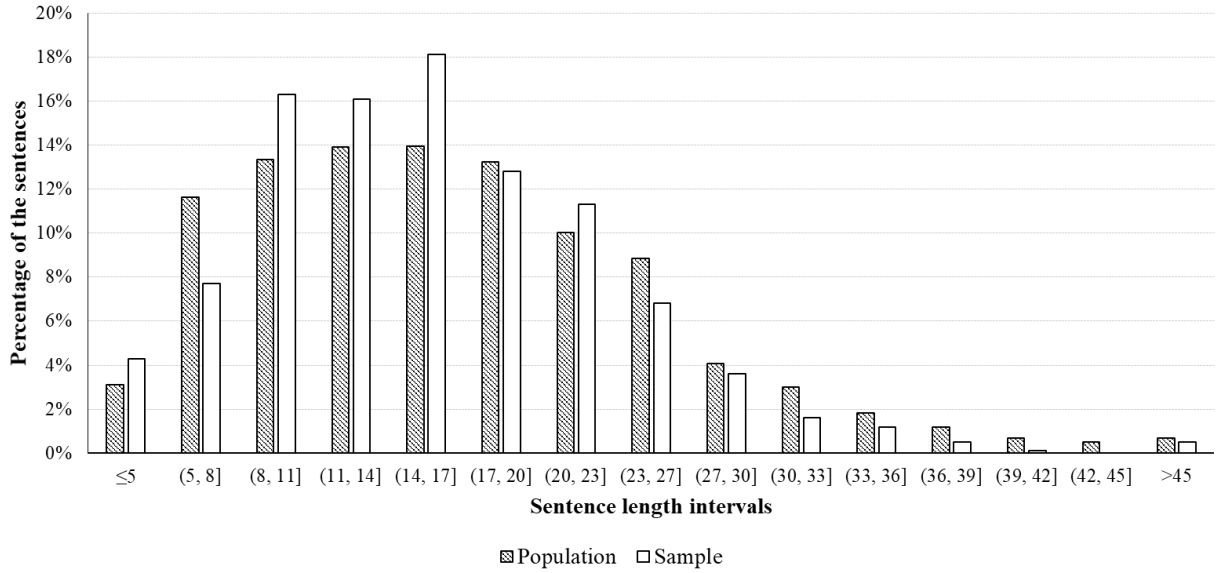


Figure 5.6. Distributions of the sentence lengths for the selected sentences (sample) and for all the sentences in the ten bridge inspection reports (population).

5.2.2.2 Feature Extraction and Representation

The training, validation, and testing configurations were first generated from the annotated training, validation, and testing sentences, respectively, using the transition-based DP model. The configuration-based features were extracted based on the defined element positions at the configurations. Second, the syntactic and semantic text features were extracted to represent these configuration-based features. The POS tag set from the Penn Treebank was used. The tags were analyzed and extracted using the commonly-used natural language tool kit (NLTK) POS tagger (Bird et al. 2009). The defined semantic classes were analyzed and extracted using the ontology-based, semi-supervised conditional random fields-based named entity recognition (NER) method (developed as per Research Task #3, in Chapter 4). In this method, the bridge deterioration knowledge ontology (developed as per Research Task #2, in Chapter 3), which represents bridge deterioration and maintenance knowledge, is used to facilitate the extraction based on content and domain-specific meaning. The errors in the semantic classes were manually checked and corrected.

Finally, the extracted text features – words, POS tags, and semantic classes – were represented by the distributed feature representations with a commonly-used vector size of 50, using the NN-based hierarchical softmax skip-gram algorithm (Mikolov et al. 2013). This algorithm was selected because it achieved the state-of-the-art performance in distributed feature representation and has been widely applied for supporting many natural language processing tasks inside (e.g., Zhou and El-Gohary 2015) and outside (e.g., Chen and Manning) of the construction domain.

5.2.2.3 Algorithm Training

The algorithm training aimed to learn the weight vectors for the constituent NN classifiers and the combiner SVM classifier. The training included three main steps. First, all the training configurations were sampled into the defined configuration clusters based on Eq. (5.1). Second, the four constituent NN classifiers were developed. Each constituent classifier corresponded to a cluster and was trained using the configurations and their GSTs of the cluster. To learn the weights for the NN classifiers [as per Eq. (5.4)], the backpropagation algorithm (Rumelhard et al. 1986) was used. It was selected because it is the workhorse of parameter learning in neural networks. Third, a combiner SVM classifier was developed. The combiner classifier was trained using all the training configurations and their GSTs. During the training, each of the configurations was represented with the probability vector (as per Figure 5.2). To learn the weight vector of the combiner SVM classifier, the stochastic gradient descent algorithm was used. This algorithm was selected because it has been widely applied in the optimization process of SVM classifier training.

5.2.2.4 Evaluation

The evaluation included algorithm validation and testing. Algorithm validation was conducted, using the configurations, to: (1) select the hyperparameter values for the classifiers, (2) select the feature representation, and (3) compare the performance of the proposed DP algorithm to those of

the three baselines – semantic single classifier-based algorithms that used an NN or SVM classifier and a semantic stacked generalization-based algorithm that used cross-validation partitioning for sampling the configurations. The selection and comparison were conducted based on configuration-based accuracy (CA), which is the ratio of the number of correctly-classified configurations to the total number of configurations, as per Eq. (5.5). Algorithm testing was conducted, using the testing sentences, to evaluate the performance of the proposed DP algorithm (with the selected hyperparameters and feature representation) in extracting dependency relations from bridge inspection reports for representing the information about bridge conditions and maintenance actions in a semantically-rich structured way. The performance was measured in terms of precision, recall, and F-1 measure, at both the SIE and SIS levels. Precision, as per Eq. (5.6), is the ratio of the number of correctly-extracted SIEs/SISs to the total number of extracted SIEs/SISs. Recall, as per Eq. (5.7), is the ratio of the number of correctly-extracted SIEs/SISs to the total number of SIEs/SISs that should be extracted. F-1 measure, as per Eq. (5.8) (Olson and Delen 2008), is the weighted harmonic mean of precision and recall. A threefold cross-validation was performed to evaluate the generalizability of the algorithm. The confidence intervals of the mean values for these measures were also calculated to evaluate the sensitivity of the performance results. The confidence intervals were calculated using Eq. (5.9) (Brookmeyer and Crowley, 1982), where $\bar{x}$ is the mean, σ is the standard deviation, n is the number of sentences or configurations in the validation or testing set, z^* is the critical value, and $z^* \frac{\sigma}{\sqrt{n}}$ is the margin of error. At 95% confidence level, z^* = 1.96. Because prediction accuracies, precisions, and recalls generally follow a normal distribution (Lu et al. 2007; Mirza et al. 2007), such a distribution was assumed and used for calculating the confidence intervals.

$$CA = \frac{number\ of\ correctly\ classified\ configurations}{number\ of\ all\ the\ configurations} \tag{5.5}$$

$$Precision = \frac{number\ of\ correctly\ extracted\ SIEs(or\ SISs)}{number\ of\ extracted\ SIEs\ (or\ SISs)} \tag{5.6}$$

$$Recall = \frac{number\ of\ correctly\ extracted\ SIE(or\ SISs)}{number\ of\ SIEs\ (or\ SISs)\ that\ should\ be\ extracted} \tag{5.7}$$

$$F\text{-}1\ measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.8}$$

$$Confidence\ interval\ (CI) = \left(\bar{x} - z^* \frac{\sigma}{\sqrt{n}}, \bar{x} + z^* \frac{\sigma}{\sqrt{n}}\right) \tag{5.9}$$

5.3 Relation Extraction Method Evaluation

5.3.1 Hyperparameter Value Selection

The hyperparameter values for the NN and SVM classifiers were selected. Because the activation and the kernel functions are especially important for the constituent NN and the combiner SVM classifiers to collectively capture the nonlinearity of the configurations, combinations of the two types of functions were tested. Four commonly-used activation functions (identity, Gaussian, hyperbolic tan, and logistic sigmoid) and four commonly-used kernel functions (linear, polynomial, radial basis function, and sigmoid) were tested, resulting in a total of 16 combinations. The selected values are summarized in Table 5.4.

Table 5.4. Hyperparameter values of the proposed and the baseline dependency parsing algorithms.

Classifier type	Hyperparameter	Value	Explanation
Neural network classifier	Number of network layers [a]	3	This value was selected based on Chen and Manning (2014), because it balances classification accuracy and computational efficiency.
	Hidden layer size [a]	200	– [c]
	Regularization parameter [a]	10^{-8}	– [c]
	Activation function	Logistic sigmoid function or hyperbolic tan function	The combination that used the logistic sigmoid activation function and the linear kernel function achieved the highest configuration-based accuracy on both the validation and testing sets, compared to the other combinations. The logistic sigmoid function was, thus, selected for the constituent neural network classifiers. Four commonly-used activation functions, including the logistic sigmoid, identity, Gaussian, and hyperbolic tan functions, were tested. The hyperbolic tan function was selected over the other three for the single neural network classifier, because it achieved the highest configuration-based accuracy on both the validation and testing sets.
Support vector machine classifier	Soft margin constant	200 or 1	A set of values, including 1 and those ranging from 20 to 300 with a step size of 20, were tested. A value of 200 for the combiner classifier and a value of 1 for the single classifier were selected to control the margin of the decision boundaries, because they achieved the highest configuration-based accuracy, on the respective validation and testing sets.
	Kernel function	Linear kernel or radial basis function kernel	The combination that used the logistic sigmoid activation function and the linear kernel function achieved the highest configuration-based accuracy on both the validation and testing sets, compared to the other combinations. The linear kernel function was, thus, selected for the combiner support vector machine classifier. Four commonly-used kernels, including the linear, polynomial, radial basis function, and sigmoid kernels, were tested. The radial basis function kernel was selected over the other three for the single support vector machine classifier, because it achieved the highest configuration-based accuracy on both the validation and testing sets.
	Degree of the polynomial kernel [b]	2	This value was selected because it is enough to capture the nonlinear relationships between features (Ben-Hur and Weston 2010).
	Coefficient of the polynomial and sigmoid kernels [b]	1	This value was selected, because it balances the influence of higher-order terms and that of lower-order terms in the polynomial and sigmoid functions and is commonly-used in practice (Ben-Hur et al. 2008).
	Gamma of the radial basis function, polynomial, and sigmoid kernels [b]	$1/n$	This value was set to the inverse of the number of features (i.e., n), which is the commonly-used value in SVM (Chang et al. 2011) to control the curvature of the decision boundaries for preventing over-fitting.

[a] The constituent and the single neural network classifiers used the same value.
[b] The combiner and the single support vector machine classifiers used the same value.
[c] The explanation follows that above.

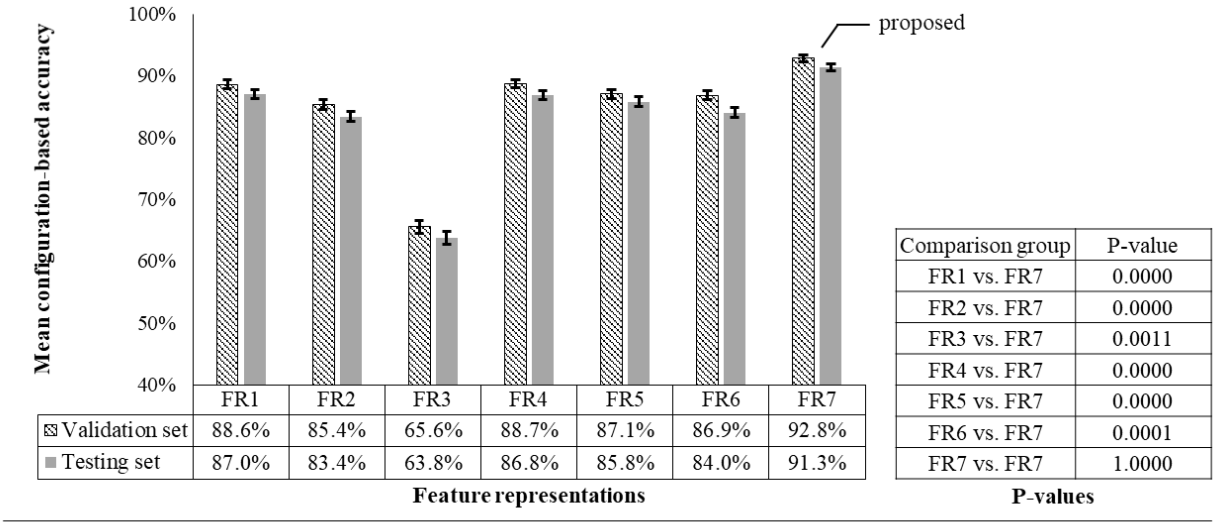
5.3.2 Feature Representation Selection

Seven text feature representations were tested and compared to investigate the effectiveness of different types of representations. These representations include combinations of the three types of features: words, POS tags, and semantic classes, as shown in Figure 5.7. To study the significance levels of the performance differences across these representations, a set of Welch's unequal variance t-tests were conducted. The probability values (p-values) were used to interpret the t-test results: if the p-value is greater than 0.05, there is no significant difference; otherwise the difference is significant. Figure 5.7 summarizes the mean configuration-based accuracies, their corresponding confidence intervals, and the p-values for comparing the proposed feature representation to the remaining six representations. The experimental results show that the proposed semantic distributed feature representation – which uses words, POS tags, and semantic classes (FR7 in Figure 5.7) – achieved the highest configuration-based accuracy of 91.3% on the testing set. By combining the high and low cooccurrence rate features, as well as the syntactic and semantic features, it was effective in capturing the highly-variable patterns of the text in the bridge inspection reports.

The feature representations (FR1, FR2, FR3, and FR6) that used words, POS tags, semantic classes, and the combination of POS tags and semantic classes achieved an accuracy of 87.0%, 83.4%, 63.8%, and 84.0%, which is 4.3%, 7.9%, 27.5%, and 7.3% lower compared to the highest (FR7), respectively, on the testing set, with all differences being significant. The lower performance was caused by two main reasons. First, the representations with a low cooccurrence rate (such as FR1) generated too many unseen feature patterns in the testing configurations that have not been learned from the training configurations. Second, the representations with a high cooccurrence rate (such as FR2, FR3, and FR6) caused the configurations that belong to different transition types to have

similar and/or identical feature patterns. The unseen and similar/identical feature patterns made the dependency parsing (DP) algorithm limited in effectively distinguishing the configurations of different transition types and, thus, resulted in a lower accuracy.

The feature representations (FR4 and FR5) that used the combination of words and POS tags and the combination of words and semantic classes achieved an accuracy of 86.8% and 85.8%, which is 4.5% and 5.5% lower compared to the highest (FR7), respectively, on the testing set, with all differences being significant. The improved performance of FR7, compared to FR4 and FR5, was mainly due to the fact that, in addition to combining the low and high cooccurrence rate features, it also utilized the POS tags and semantic classes jointly. These two types of features are complementary to each other and, thus, led to the optimal DP performance. The semantic class features are effective in capturing the dependency relations (word-to-word interactions) between the concepts that have defined semantics. For example, the words “severe” and “corrosion” can be classified into a correct transition type based on their defined semantic meanings: a categorical severity measure (“severe”, as a modifier) describes a deficiency (“corrosion”, as a head). On the other hand, the POS tag features are effective in capturing the relations between the concepts that do not have defined semantics (low-content-bearing words, such as “of”, “on”, and “at”). For example, the SIEs “wearing surface” (as a head) and “concrete deck” (as a modifier) in the phrase “wearing surface on the concrete deck” can be associated with a correct SIE-to-SIE dependency relation based on the POS tag of the “on” (i.e., preposition).



	FR1	FR2	FR3	FR4	FR5	FR6	FR7
Validation set	88.6%	85.4%	65.6%	88.7%	87.1%	86.9%	92.8%
Testing set	87.0%	83.4%	63.8%	86.8%	85.8%	84.0%	91.3%

Feature representations

Comparison group	P-value
FR1 vs. FR7	0.0000
FR2 vs. FR7	0.0000
FR3 vs. FR7	0.0011
FR4 vs. FR7	0.0000
FR5 vs. FR7	0.0000
FR6 vs. FR7	0.0001
FR7 vs. FR7	1.0000

P-values

- FR1 = words; FR2 = POS tags; FR3 = semantic classes; FR4 = words + POS tags; FR5 = words + semantic classes; FR6 = POS tags + semantic classes; FR7 = words + POS tags + semantic classes.
- The "I-shape" bar indicates the confidence interval calculated from the corresponding mean configuration-based accuracy.
- The p-values were calculated from the Welch's unequal variances t-tests (using the testing set) and are significant at 0.05 level (2-tailed).

Figure 5.7. Performance results for feature representation selection.

5.3.3 Comparison to Baseline Algorithms

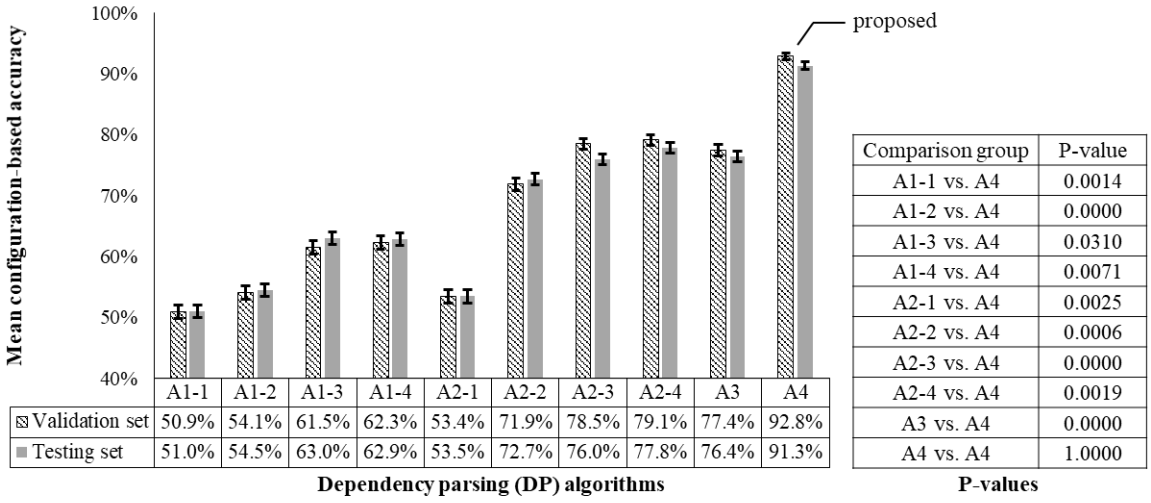
Three baseline DP algorithms were developed for comparative evaluation: a semantic NN-based, a semantic SVM-based, and a semantic stacked generalization (SG)-based. The first two were used to evaluate the effectiveness of the proposed ensemble learning-based approach. The semantic NN-based DP baseline was selected because it is one of the state-of-the-art NN-based DP methods (e.g., Chen and Manning 2014) that has been commonly used as a benchmark (e.g., by Weiss et al. 2015; Alberti et al. 2015). The semantic SVM-based DP baseline was selected because it is commonly used in the literature (e.g., Kudo and Matsumoto 2002; Yamada and Matsumoto 2003). For these baselines, a single NN or SVM classifier was used. The hyperparameter values of the classifiers are shown in Table 5.4. The third baseline was used to evaluate the effectiveness of the proposed sampling approach. It was selected because it is the most similar to the proposed algorithm – except that the proposed algorithm used the similarity-based sampling method rather

than cross-validation partitioning. For the SG-based and the proposed DP algorithms, four constituent NN classifiers with the logistics sigmoid activation function and a combiner SVM classifier with the linear kernel functions were used. These functions were selected based on the results in Section 5.3.1. All the DP algorithms (for both the proposed and the baseline models) were developed using the proposed semantic distributed feature representation, as per Figure 5.2.

The performances of the proposed and the baseline algorithms are summarized in Figure 5.8, and their confusion matrices are shown in Figure 5.9. As shown in Figure 5.8, the proposed semantic NNE-based DP algorithm (A4) achieved the highest accuracy of 91.3% on the testing set. The semantic NN-based DP baseline with the hyperbolic tan activation function (A2-4) achieved an accuracy of 77.8%. The semantic SVM-based DP baseline with the radial basis function kernel (A1-3) achieved an accuracy of 63.0%. And, the semantic SG-based DP algorithm (A3) achieved an accuracy of 76.4%. As shown in Figure 5.9, the proposed algorithm achieved improved precisions and recalls across all transition types.

These results indicate that the proposed ensemble learning-based DP approach is effective in dealing with highly technical, domain-specific text (such as that in the bridge inspection reports) for extracting dependency relations. It significantly improved the accuracy by 13.5% and 28.3%, compared to the NN- and SVM-based DP baselines, respectively. This is because, by using the constituent and combiner classifiers, the proposed ensemble learning-based DP approach was able to sufficiently capture the distributions of all the configurations, which were too complex to be captured by a single classifier. The results also indicate that the proposed similarity-based sampling method is effective in capturing the complex configuration distributions of the text. It significantly improved the accuracy by 14.9% compared to the SG-based DP baseline. This is because the sampling method used the similarities measured in multiple feature spaces as a

collective criterion to sample the configurations into meaningful clusters (see Section 5.2.1.2). Conversely, the SG-based algorithm simply clustered the configurations using cross-validation partitioning. Only learning from the similarly-distributed and more easily-separable configurations allowed each constituent classifier to sufficiently capture the local distributions of the configurations, which resulted in more effective ensembling – improved ability to capture the global distributions of all the configurations.



	A1-1	A1-2	A1-3	A1-4	A2-1	A2-2	A2-3	A2-4	A3	A4
Validation set	50.9%	54.1%	61.5%	62.3%	53.4%	71.9%	78.5%	79.1%	77.4%	92.8%
Testing set	51.0%	54.5%	63.0%	62.9%	53.5%	72.7%	76.0%	77.8%	76.4%	91.3%

Dependency parsing (DP) algorithms

Comparison group	P-value
A1-1 vs. A4	0.0014
A1-2 vs. A4	0.0000
A1-3 vs. A4	0.0310
A1-4 vs. A4	0.0071
A2-1 vs. A4	0.0025
A2-2 vs. A4	0.0006
A2-3 vs. A4	0.0000
A2-4 vs. A4	0.0019
A3 vs. A4	0.0000
A4 vs. A4	1.0000

P-values

- A1-1, A1-2, A1-3, and A1-4 are semantic support vector machine-based DP algorithms with linear, polynomial, radial basis function (RBF), and sigmoid kernel functions, respectively.
- A2-1, A2-2, A2-3, and A2-4 are semantic neural network-based DP algorithms with Gaussian, identity, logistic sigmoid, and hyperbolic tan activation functions, respectively.
- A3 is the semantic stacked generalization-based DP algorithm; A4 is the proposed semantic neural network ensemble-based DP algorithm.
- The "I-shape" bar indicates the confidence interval calculated from the corresponding mean configuration-based accuracy.
- The p-values were calculated from the Welch's unequal variances t-tests (using the testing set) and are significant at 0.05 level (2-tailed).

Figure 5.8. Performances of different dependecy parsing (DP) algorithms.

Gold standard transition type

Predicted transition type	Shift	Left-arc	Right-arc	Precision
Shift	2497	1283	323	60.9%
Left-arc	374	1787	145	77.5%
Right-arc	240	676	897	49.5%
Recall	80.3%	47.7%	65.7%	**63.0%**

(a) confusion matrix for semantic SVM-based dependency parsing algorithm.

Gold standard transition type

Predicted transition type	Shift	Left-arc	Right-arc	Precision
Shift	3433	370	300	83.7%
Left-arc	423	1754	129	76.1%
Right-arc	436	169	1208	66.6%
Recall	80.0%	76.5%	73.8%	**77.8%**

(b) confusion matrix for semantic NN-based dependency parsing algorithm.

Gold standard transition type

Predicted transition type	Shift	Left-arc	Right-arc	Precision
Shift	3391	458	262	82.6%
Left-arc	369	1823	115	79.1%
Right-arc	560	175	1069	59.0%
Recall	78.4%	74.2%	74.1%	**76.4%**

(c) confusion matrix for semantic SG-based dependency parsing algorithm.

Gold standard transition type

Predicted transition type	Shift	Left-arc	Right-arc	Precision
Shift	3819	153	131	93.1%
Left-arc	112	2132	62	92.5%
Right-arc	165	93	1555	85.8%
Recall	93.2%	89.7%	89.0%	**91.3%**

(d) confusion matrix for semantic NNE-based dependency parsing algorithm (proposed).

- The bold font indicates the mean configuration-based accuracy.
- The precision is the number of correctly-classified transitions of a type out of the number of transitions classified as the type.
- The recall is the number of correctly-classified transitions of a type out of the number of gold standard transitions of the type.
- The precisions, recalls, and accuracies were calculated using the testing set.
- SVM = support vector machine; NN = neural network; SG = stacked generalization; NNE = neural network ensemble.

Figure 5.9. Confusion matrices for the proposed and the baseline dependency parsing algorithms.

5.3.4 Performance of the Proposed Dependency Parsing Algorithm

The performance of the proposed semantic NNE-based DP algorithm was evaluated in extracting dependency relations from bridge inspection reports for representing the information about bridge conditions and maintenance actions into SIEs and SISs. The SIE-level measures evaluated how well the individual SIEs can be correctly represented, while the SIS-level measures evaluated how well the SISs can be correctly represented (an SIS representation is correct if and only if all its constituent SIEs are represented correctly). The SIS-level measures are, thus, more stringent compared to the SIE-level measures. Examples of the extracted information are provided in Figure 5.10. The experimental results are summarized in Figure 5.11.

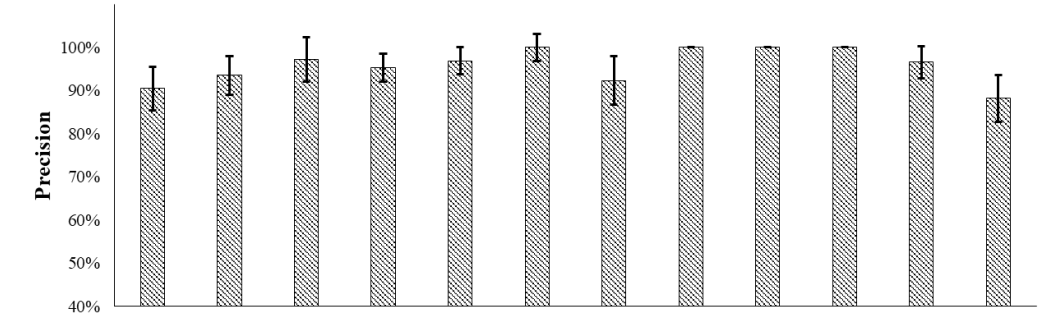
Sentence #1	The underside of cap beam between columns C1 and C2, exhibits hollow sounding concrete areas up to 44 " x 36 " hollow sounding concrete, 18 " x 12 " x 3 " deep spall and 2 ' x 8 " x 3 " deep spall with exposed rebars.									
Extraction	**Semantic information set (SIS)**									
	ET	**DY**	**DC**	**NM**	**NU**	**QM**	**SM**	**MA**	**MM**	**DT**
	cap_beam	hollow_sounding_concrete	–	44_36	"_"	–	–	–	–	–
	cap_beam	hollow_sounding_concrete	–	–	–	–	–	–	–	–
	cap_beam	spall	–	18_12_3	"_"_"	–	deep	–	–	–
	cap_beam	spall	–	2_8_3	'_"_"	–	deep	–	–	–
	cap_beam	exposed_rebars	–	2_8_3	'_"_"	–	deep	–	–	–

Sentence #2	The timber deck members were coated with creosote and tar.									
Extraction	**Semantic information set (SIS)**									
	ET	**DY**	**DC**	**NM**	**NU**	**QM**	**SM**	**MA**	**MM**	**DT**
	timber_deck	–	–	–	–	–	–	coated	creosote	–
	timber_deck	–	–	–	–	–	–	coated	tar	–

Sentence #3	Truss diagonals and verticals typically have corrosion and section loss at the lower gusset connections.									
Extraction	**Semantic information set (SIS)**									
	ET	**DY**	**DC**	**NM**	**NU**	**QM**	**SM**	**MA**	**MM**	**DT**
	truss_diagonals	–	corrosion	–	–	–	–	–	–	–
	truss_diagonals	–	section_loss	–	–	–	–	–	–	–
	truss_verticals	–	corrosion	–	–	–	–	–	–	–
	truss_verticals	–	section_loss	–	–	–	–	–	–	–

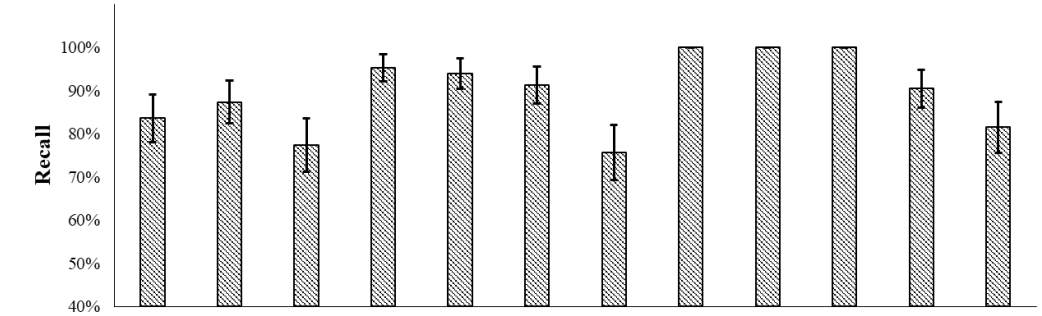
ET = bridge element; DY = deficiency; DC = deficiency cause; NM = numerical measure; NU = numerical measure unit; QM = categorical quantity measure; SM = categorical severity measure; MA = maintenance action; MM = maintenance material; DT = date.

Figure 5.10. The semantic information element (SIE) level and the semantic information set (SIS) level performance of the proposed algorithm.



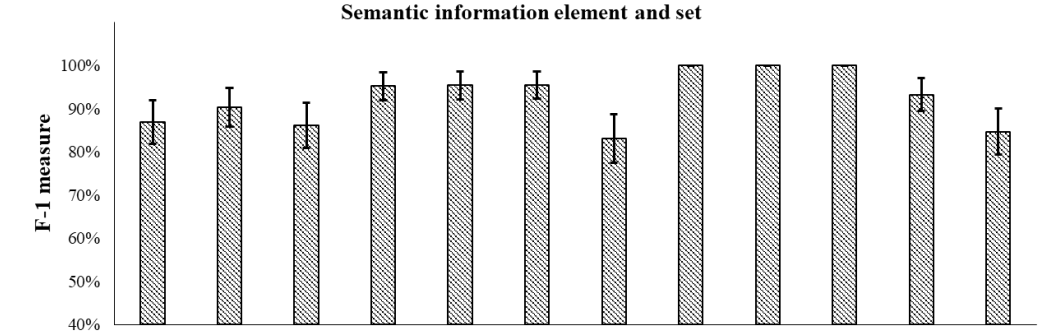
	ET	DY	DC	NM	NU	QM	SM	MA	MM	DT	Avg.	SIS
Prec. (%)	90.5	93.5	97.2	95.2	96.9	100.0	92.3	100.0	100.0	100.0	96.6	88.2
MOE (%)	±5.1	±4.4	±5.2	±3.2	±3.1	±3.1	±5.6	±0.0	±0.0	±0.0	±3.8	±5.4

Semantic information element and set



	ET	DY	DC	NM	NU	QM	SM	MA	MM	DT	Avg.	SIS
Rec. (%)	83.6	87.3	77.4	95.2	93.9	91.3	75.7	100.0	100.0	100.0	90.4	81.5
MOE (%)	±5.5	±5.0	±6.3	±3.2	±3.6	±4.2	±6.4	±0.0	±0.0	±0.0	±4.4	±5.8

Semantic information element and set



	ET	DY	DC	NM	NU	QM	SM	MA	MM	DT	Avg.	SIS
F-1 (%)	86.9	90.3	86.1	95.2	95.4	95.5	83.1	100.0	100.0	100.0	93.3	84.7
MOE (%)	±5.1	±4.4	±5.2	±3.2	±3.1	±3.1	±5.6	±0.0	±0.0	±0.0	±3.8	±5.4

Semantic information element and set

- ET = bridge element; DY = deficiency; DC = deficiency cause; NM = numerical measure; NU = numerical measure unit; QM = categorical quantity measure; SM = categorical severity measure; MA = maintenance action; MM = maintenance material; DT = date.
- Avg. = average semantic information element level performance; SIS = semantic information set level performance.
- Prec. = precision; Rec. = Recall; F-1 = F-1 measure; MOE = margin of error, where a confidence interval = (mean - MOE, mean + MOE).

Figure 5.11. The semantic information element (SIE) level and the semantic information set (SIS) level performance of the proposed algorithm.

5.3.4.1 Performance at the Semantic Information Element Level

At the SIE level, on average, the proposed semantic NNE-based DP algorithm achieved a precision, recall, and F-1 measure of 96.6%, 90.4%, and 93.3%, respectively. For some SIE types (e.g., ET, DY, DC, and SM), the algorithm achieved results lower than these averages. For the ET and DY SIEs, it achieved an SIE-level precision, recall, and F-1 measure of 90.5%, 83.6% and 86.9%, and 93.5%, 87.3%, and 90.3%, which are 6.1%, 6.8%, and 6.4%, and 3.1%, 3.1%, and 3.0% lower compared to the averages, respectively. Two main sources of errors that contributed to these results were identified. First, the large number of the ET and DY SIEs negatively affected the performance of the algorithm. Bridge inspection reports tend to have more descriptions about bridge elements and their deficiencies. For example, in the used dataset, 50.2% and 22.1% of the concepts are ET and DY SIEs, respectively. These SIEs are, thus, the main sources of the ambiguities in the dependency relations (i.e., associating the right DY elements to the right ET elements is challenging, given the existences of multiple such SIEs in a sentence). Second, the errors generated during the POS tagging process negatively affected the performance of the algorithm. For example, in the following sentence, the words "shows" and "cut" were incorrectly tagged as "noun" and "verb", respectively: "The salvaged stringer superstructure, shows flame cut holes for various stringer ends, over the east abutment, and along the fascia stringer ends, over each side, of the pier." (CDOT 2009). This resulted in incorrectly associating the DY element ("flame cut holes") to the other ET elements (e.g., "east abutment", "fascia stringer ends", and "pier"), instead of correctly associating it to "stringer superstructure".

For the DC and SM SIEs, the algorithm achieved an SIE-level precision, recall, and F-1 measure of 97.2%, 77.4%, and 86.1%; and 92.3%, 75.7%, and 83.1%, respectively. The recalls of these two SIEs are much lower than the average (13.0% and 14.7% lower for DC and SM, respectively).

Two main sources of errors that caused the lower recalls were identified. First, when combining words into SIEs, the information representation method (as per Figure 5.4, step 2) sometimes combined multiple DC SIEs into one single element, and thus led to the low recall for the DC SIEs (multiple DC SIEs should be extracted, while only the single DC element was incorrectly extracted). For example, in the following sentence, the DC SIEs "rodent droppings" and "debris" were combined into one DC element "rodent droppings debris", which is incorrect: "The interior of the longitudinal box girders exhibited heavy rodent droppings, debris, and nests…." (LaDOTD 2008). For a correct extraction and representation, the three DC SIEs "rodent droppings", "debris", and "nests" should all be extracted and represented as separate SIEs. A further analysis revealed the root source of such mistakes: ignoring punctuation during the parsing (which is the default practice according to the universal dependencies guideline (Marneffe et al. 2014) and is commonly applied in other DP methods). Punctuations are in some cases indicative of correct dependency relations. So, when the comma between "rodent droppings" and "debris" was not considered, they were associated with an incorrect dependency relation. Second, the proposed SIE-to-SIE dependency relation types (as per Figure 5.3) sometimes limited the information representation, and thus led to the lower recall of the SM SIEs. For example, in the sentence above, the SM SIE "heavy" and the DC SIE "rodent droppings" should be extracted and represented. Although the DP algorithm correctly associated a dependency relation between these two elements, the SM SIE was not represented in an SIS because no semantics (SIE-to-SIE dependency relation types) were defined between the SM and DC SIEs.

5.3.4.2 Performance at the Semantic Information Set Level

At the SIS level, the proposed semantic NNE-based DP algorithm achieved a precision, recall, and F-1 measure of 88.2%, 81.5%, and 84.7%, respectively. Compared to the average SIE-level

measures, the SIS-level precision, recall, and F-1 measure are 8.4%, 8.9%, and 8.6% lower, respectively. This is because (as discussed) the SIS-level measures are naturally more stringent than the SIE-level ones. The results also show that the performance in extracting/representing the bridge elements (at the SIE level) sets an upper bound for the entire SIS-level performance. This is because the bridge elements are the root of the extraction and representation, so when a bridge element is extracted and represented incorrectly, its whole SIS becomes incorrect.

CHAPTER 6 – UNSUPERVISED DATA LINKING

This chapter presents the proposed data linking method for linking data records that are extracted from textual bridge inspection reports (extracted as per Research Tasks #3 and #4). The method development and evaluation (Research Task #5) are presented in this chapter.

6.1 Comparison to the State of the Art

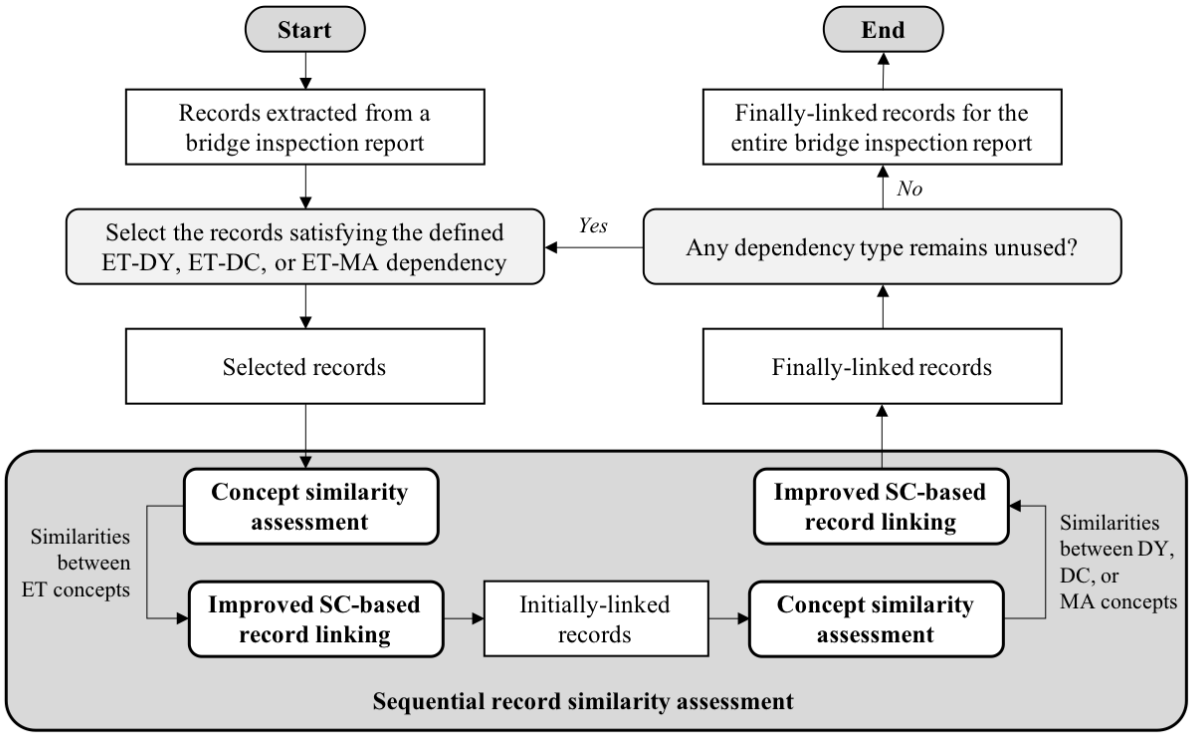
Generally, data linking includes three primary components: (1) attribute similarity assessment: assessing the similarities between the corresponding attributes of the records, (2) record similarity assessment: assessing the similarities between the records based on their attribute similarities, and (3) record linking: linking the records based on their similarities. Many data linking methods have been developed in various application domains, such as healthcare (Brook et al. 2008), national security and crime investigation (Phua et al. 2012), government service (Winkler 2006), etc. Despite the importance of existing methods, they are still limited in linking data extracted from highly technical, domain-specific documents, such as bridge inspection reports. First, most of the existing methods (e.g., Fu et al. 2014; Fisher et al. 2015; Karapiperis and Verykios 2014; Karapiperis and Verykios 2015; De Leone and Minnetti 2015) use term-level string comparisons to assess the similarities between simple entities (e.g., numbers and single terms). Such methods are rather insufficient in assessing the similarities between more complex entities, such as those extracted from bridge inspection reports (e.g., domain-specific concepts). Second, most of the existing methods use vector representations of attribute similarities to assess record similarity (Vatsalan et al. 2013). Such methods are limited in effectively assessing the similarities of records when dependencies among attribute similarity assessments exist (Ananthakrishna 2002; Weis and Naumann 2004). Although a limited number of methods (e.g., Weis and Naumann 2004; Albrech and Naumman 2008; Puhlmann et al. 2006) have been developed for addressing this limitation for

domain-general applications, they cannot be used for capturing domain-specific dependencies, such as those carried in the bridge report records. Third, the majority of existing data linking methods are classification-based (Vatsalan et al. 2013), where pairwise classifications are conducted for linking record pairs. Such methods open the door to transitive closures, which typically lead to false positives (Christen 2012; Elmagarmid et al. 2007).

6.2 Data Linking Method Development

6.2.1 Proposed Data Linking Method

The proposed data linking method is composed of three primary sub-methods: concept similarity assessment, record similarity assessment, and SC-based record linking. An overview of the proposed method is presented in Figure 6.1.



Note: ET = bridge element; DY = deficiency; DC = deficiency cause; MA = maintenance action; SC = spectral clustering.

Figure 6.1. Overview of the proposed spectral clustering (SC)-based data linking method.

6.2.1.1 Concept Similarity Assessment

A new concept similarity (CS) assessment method was proposed, which assesses the similarities between the concepts based on the similarity degrees of their terms. Three alternative CS scoring functions were proposed and tested: term-based, relative-position-based, and right-position-based functions. For all three functions, the assumption is: at different places in a report, the writer could use different terminologies to refer to the same entity/instance. For example, although linguistically "west longitudinal box girder" is a subconcept of "longitudinal girder", the writer of the inspection report used both terminologies at different places to refer to the same girder; and the variability in terminology was merely an inconsistency in writing style.

The term-based function is based on the following hypothesis: the similarity of two concepts is best assessed based on the similarity degrees of their most-similar terms. For each concept pair, this function (1) assesses the string-based term similarity (TS) of each term in the shorter concept x to each term in the longer concept y, using an existing TS scoring function, (2) selects the most-similar term from concept y – the one with the highest string-based TS score – for each term in concept x, and (3) uses the normalized total of the selected scores as the concept similarity degree. The term-based CS score is calculated using Eq. (6.1), where $CS(x, y)$ is the concept similarity of concept x to concept y; t_i and t_j are the terms of concepts x and y, respectively; $TS(t_i, t_j)$ is the string-based term similarity of t_i to t_j; and n and m are the lengths of concepts x and y, respectively.

$$Term\text{-}based\ CS\ Score = CS(x, y) = \frac{1}{n} \times \sum_{i=1}^{n} \max_{1 \le j \le m} \{TS(t_i, t_j)\} \quad (6.1)$$

The relative-position-based function follows the same hypothesis and steps of the term-based function. But, in addition to the string-based term similarity, it also considers the position-based term similarity when selecting the most-similar terms, because the positions of terms in concepts can also capture the term similarity. The position-based similarity between a pair of terms is calculated using a relative position score, which is 1 minus the absolute difference between the normalized position of t_i in concept x (i.e., i/n) and that of t_j in concept y. The relative position scores are multiplied with the string-based TS scores for selecting the most-similar terms. The relative-position-based CS score is calculated using Eq. (6.2), where i and j are the positions of t_i and t_j in concepts x and y, respectively; i/n and j/m are the normalized positions of t_i and t_j, respectively; $1 - |i/n - j/m|$ is the relative position score between t_i and t_j; and the other notations follow those defined in Eq. (6.1).

$$Relative\text{-}position\text{-}based\ CS\ Score = CS(x, y)$$
$$= \frac{1}{n} \times \sum_{i=1}^{n} \underset{1 \le j \le m}{max} \left\{ TS(t_i, t_j) \times \left(1 - \left|\frac{i}{n} - \frac{j}{m}\right|\right)\right\} \quad (6.2)$$

The right-position-based function is based on the following hypothesis: in a multi-term concept name, the contribution of a term's meaning to the concept meaning decreases from right to left; the most right term (i.e., the last term) contributes the most (Zhang and El-Gohary 2016). Thus, in addition to the string-based term similarity, this function also considers the contribution level of the term meaning to the concept meaning. The contribution level is calculated using a right position score, which is the ratio between the summed position indices of the paired terms and that of all the paired terms. As such, for each concept pair, this function (1) assesses the string-based TS of each term in concept x to each term in concept y, (2) adjusts the string-based TS score of each term pair by multiplying it with its right position score, and (3) uses the total of all the adjusted

scores as the concept similarity degree. The right-position-based CS score is calculated using Eq. (6.3), where $(i+j)/\sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)$ is the right position score and the other notations follow those defined in Eq. (6.1). Examples of using these functions to assess the similarity between two concepts are provided in Figure 6.2.

$$Right\text{-}position\text{-}based\ CS\ Score\ = CS(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(TS(T_i,T_j)\times\frac{(i+j)}{\sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)}\right) \quad (6.3)$$

		Concept B			
		west	longitudinal	box	girder
Concept A	longitudinal	0.083	**1.000**	0.083	0.250
	girder	0.000	0.250	0.000	**1.000**

Term-based concept similarity score ("longitudinal girder", "west longitudinal box girder") = 1/2 x (1.000 + 1.000) = 1.000

(a) Term-based concept similarity scoring function

		Concept B			
		west	longitudinal	box	girder
Concept A	longitudinal	0.062	**1.000**	0.062	0.125
	girder	0.000	0.125	0.000	**1.000**

Relative-position-based concept similarity score ("longitudinal girder", "west longitudinal box girder") = 1/2 x (1.000 + 1.000) = 1.000

(b) Relative-position-based concept similarity scoring function

		Concept B			
		west	longitudinal	box	girder
Concept A	longitudinal	**0.005**	**0.094**	**0.010**	**0.039**
	girder	**0.000**	**0.031**	**0.000**	**0.188**

Right-position-based concept similarity score ("longitudinal girder", "west longitudinal box girder") = (0.005 + 0.094 + …...) = 0.367

(c) Right-position-based concept similarity scoring function

Note: *The numbers in bold font were selected in the corresponding function for calculating the concept similarity.*

Figure 6.2. Examples for the proposed concept similarity (CS) scoring functions. (a) The term-based CS scoring function. (b) The relative-position-based CS scoring function. (c) The right-position-based CS scoring function.

6.2.1.2 Record Similarity Assessment

A new sequential record similarity assessment method was proposed, which breaks down the record-level similarity assessment task into sequences of attribute-level tasks based on similarity assessment dependencies. Three types of similarity assessment dependencies were defined and used: element-deficiency, element-deficiency cause, and element-maintenance action.

The element-deficiency (ET-DY) dependency is used to break down the record assessment task for the records that include bridge element (ET) and deficiency (DY) attribute values (i.e., concept names in this case). This dependency assumes that: (1) if two records in a bridge inspection report have same/similar ET and DY concepts, then they refer to the same deficiency instance, i.e., the same deficiency on the same element of the same bridge (e.g., a specific "crack" on a specific "timber deck" of a specific "bridge"); and (2) for the same deficiency instance, its deficiency characteristics (numerical measure, categorical quantity measure, and categorical severity measure) as well as maintenance action and material should be the same. Under this dependency, the assessment of record similarity is conducted, sequentially, based on the ET and DY concepts. First, the similarities of the records are assessed based on that of the ET concepts. Second, for the records including same/similar ET concepts, their similarities are further assessed based that of the DY concepts.

The element-deficiency cause (ET-DC) dependency is used to break down the record assessment task for the records that include ET and deficiency cause (DC) attribute values, but no DY attribute values. This dependency assumes that: (1) if two records in a bridge inspection report have same/similar ET and DC concepts, then they refer to the same deficiency cause instance, i.e., the same deficiency cause on the same element of the same bridge (e.g., a specific "white rot" on a specific "timber deck" of a specific "bridge"); and (2) for the same deficiency cause instance, its

maintenance action and material should be the same. Under this dependency, the assessment of record similarity is conducted, sequentially, based on the ET and DC concepts.

The element-maintenance action (ET-MA) dependency is used to break down the record assessment task for the records that include ET and maintenance action (MA) attribute values, but no DY and no DC attribute values. This dependency assumes that (1) if the records in a bridge inspection report have same/similar ET and MA concepts, then they refer to the same maintenance action instance, i.e., the same action performed on the same element of the same bridge (e.g., a specific "splicing augmentation" performed on a specific "timber beam" of a specific "bridge"); and (2) for the same maintenance action instance, the material used should be the same. Under this dependency, the assessment of record similarity is conducted, sequentially, based on the ET and MA concepts.

6.2.1.3 Spectral Clustering-Based Record Linking

An improved SC-based data linking method was proposed to link the records in an unsupervised way at each attribute level, without forming transitive closures. Figure 6.3 shows an overview of the proposed data linking method.

The improved spectral clustering uses a proposed iterative bi-partitioning method to automatically identify the optimal number of target clusters for data linking, without using a manually pre-defined number. The bi-partitioning includes three main steps. First, the original NJW normalized SC method (Ng et al. 2002) is used to always bi-partition a parent cluster into two child clusters, where the parent cluster contains concepts to be linked and each of the child clusters contains linked concepts. Second, the quality of the partitioning is assessed by a proposed partitioning quality assessment function. Third, the assessment score is compared to a user-defined threshold.

If the score is greater than the threshold, the original parent cluster is eliminated, and the children become the new parent clusters, so that the linked concepts in each child cluster are further partitioned. Otherwise, the resulting child clusters are eliminated, and the original parent cluster is kept without further partitioning. These steps are repeated until no more parent clusters can be partitioned.

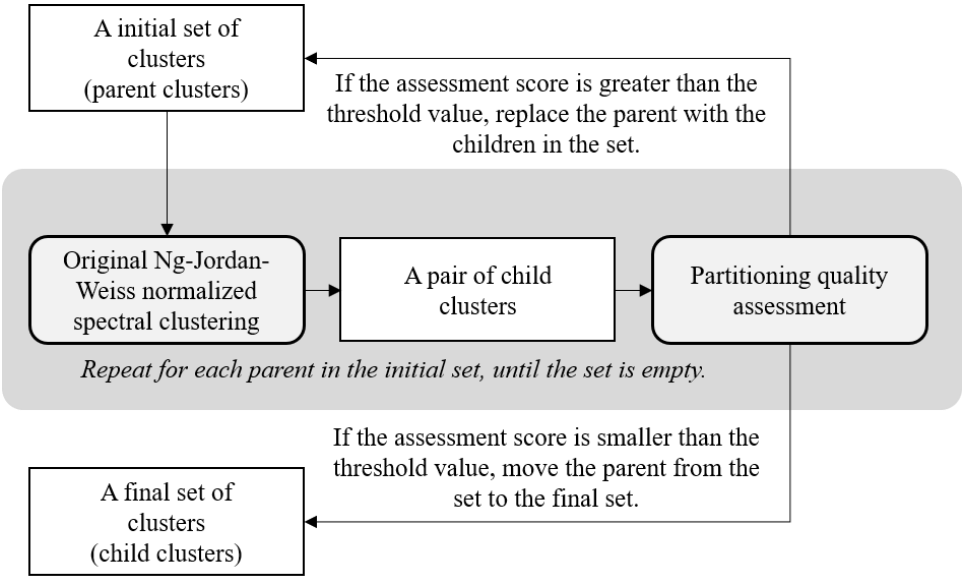


Figure 6.3. The improved spectral clustering (SC)-based data linking method.

The partitioning quality assessment function was proposed based on the following assumption: a high-quality partitioning should result in a high inter-cluster distance and a low intra-cluster distance. The function is defined in Eq. (6.4), where *PAQ score* is a partitioning quality assessment score; PC is a parent cluster; C^i and C^j are concepts in child clusters CC_1 and CC_2, respectively; C^k and C^l are concepts in the same child cluster; $CS(C^i, C^j)$ is the concept similarity between C^i and C^j; NB is the number of unique concept pairs between CC_1 and CC_2, which is used for normalizing the total inter-cluster distance; and NW is the total number of unique concept pairs in each of the child clusters, which is used for normalizing the total intra-cluster distance. The

function considers two cases. First, when a parent cluster only contains two concepts (i.e., each of its children has one concept), the function uses the ratio between the dissimilarity and the similarity to capture the partitioning quality. It assigns a high PQA score to a high-quality partitioning, which in this case indicates higher dissimilarity and lower similarity. Second, when the parent cluster contains more than two concepts, the function uses a revised McClain-Rao index (McClain and Rao 1975) to capture the partitioning quality. It assigns a high PQA score to a high-quality partitioning, which in this case indicates higher average inter-cluster distance and lower intra-cluster distance.

$$PQA\ score = \begin{cases} \dfrac{1 - CSAF(i,j)}{CSAF(i,j)} & if\ |PC| = 2; \\ \dfrac{\sum_{i \in CC_1} \sum_{j \in CC_2} 1 - CSAF(i,j)}{\sum_{CC \in \{CC_1, CC_2\}} \sum_{k,l \in CC; k<l} 1 - CSAF(k,l)} \times \dfrac{NB}{NW} & Otherwise. \end{cases} \quad (6.4)$$

Since the spectral clustering represents concepts using similarity graphs, the use of unsupervised pre-classification prior to the clustering – to break down a similarity graph into several small ones – was tested to evaluate if the size reduction of the graph would improve the performance of the clustering. The pre-classification was formulated as a linear sum assignment optimization problem, which aims to classify a pair of concepts into "match" and "non-match" (a "match" means that the concepts should be linked). A constraint that a concept cannot be linked to itself was added to the original optimization problem to avoid having all concepts only linked to themselves. The linked concept pairs were grouped to form small graphs/clusters. For example, if concept *x* is linked to concept *y* and concept *y* is linked to concept *z*, the three concepts are grouped into a cluster.

6.2.2 Implementation of the Proposed Method

The proposed data linking method was implemented in a Python program. The program includes two modules: a record preprocessing module and a data linking module. The preprocessing module conducts morphological analysis to map several variants of a term into a single root form, thereby facilitating the assessment of term similarity. For example, it mapped "cracks" and "cracked" into the same root "crack". The natural language toolkit (NLTK) Porter stemmer (Bird et al. 2009) was used for conducting the morphological analysis. The linking module links the preprocessed records. The concept similarity assessment and the iterative bi-partitioning methods were implemented in Python. The Freely Extensible Biomedical Record Linkage (FEBRL) package (Christen 2008) was used for term similarity assessment. The Source Scientific Tools for Python (SciPy) package (Jones et al. 20001) was used for unsupervised pre-classification. The Warshall's algorithm (Cormen et al. 1990) was used for grouping concept pairs into small clusters.

6.3 Data Linking Method Evaluation

A set of experiments were conducted to test and evaluate the performances of the data linking method and its sub-methods. A total of 1,743 records (extracted from ten bridge inspection reports) were automatically linked using the proposed data linking algorithm, and the linking results were evaluated based on precision and recall. Five main experiments were conducted to test: (1) the performances of the term similarity scoring functions, (2) the performances of the concept similarity scoring functions, (3) the performance of the sequential record similarity assessment method, (4) the performance of the improved SC-based data linking method, and (5) the overall performance of the proposed data linking method. Figure 6.4 shows examples of the records linked by the proposed method.

No.	Examples of linked records extracted from bridge inspection reports									
	ET	DY	DC	MA	MM	NM	NU	QM	SM	DT
1	Deck	Crack	–	–	–	–	–	–	–	2009
	Deck	Cracks	–	–	–	–, 4, –, –	FT, –, –, –	–	–	–
	Deck	Cracks	–	–	–	–, 4, –, –	FT, –, –, –	–	–	–
	Concrete deck	Cracks	–	–	–	–	–	–	–	2015
	Concrete deck	Cracks	–	–	–	–	–	–	Moderate	–
	Concrete deck	Cracks	–	–	–	–	–	–	–	2015
2	Bridge deck	Delamination	–	–	–	–	–	–	–	–
	Bridge deck	Delamination	–	–	–	–	–	–	–	2011
3	Truss bearings	Surface rust	–	–	–	–	–	–	–	–
	Bearings	Surface rust	–	–	–	–	–	–	–	1997
	Rollernest bearing assemblies	Surface rust	–	–	–	–	–	–	–	–
	Bearing	Surface rust	–	–	–	–	–	–	–	1997
	Bearings	Surface rust corrosion	–	–	–	–	–	–	–	2000
4	Masonry plates	Undermined	–	–	–	–, –, 1.25, –	–, –, IN, –	–	–	–
	Masonry plates	Undermined	–	–	–	–, –, 1, –	–, –, IN, –	–	–	–
	South bearing masonry plate	Undermined	–	–	–	–, –, 0.25, –	–, –, IN, –	–	–	–
	Bearing masonry plates	Undermined	–	–	–	30, –, 2, –	IN, –, IN, –	–	–	–
5	Seals	–	Accumulation debris	–	–	–	–	–	Heavy	–
	Seals	–	Accumulation of sand dirt debris	–	–	–	–	–	Heavy	–
	Seal	–	Accumulation of sand	–	–	–	–	–	–	–
	Seal	–	Accumulation of dirt debris	–	–	–	–	–	–	–
6	West abutment	Void undermined area	–	–	–	8, 0.5, 0.5, –	FT, FT, FT, –	–	–	–
	West abutment	Void undermined area	–	–	–	8, 5, 0.5, –	FT, FT, FT, –	–	–	–
	West abutment	Void undermined area	–	–	–	8, 5, 0.5, –	FT, FT, FT, –	–	–	–
	Abutment	Void undermining	–	–	–	8, 0.5, 0.6, –	FT, FT, FT, –	–	–	–
	Abutment	Void undermining	–	–	–	8, 0.5, 0.5, –	FT, FT, FT, –	–	–	–

- ET, DY, DC, MA, MM, NM, NU, QM, SM and DT are the record attributes, where ET = bridge element, DY = deficiency, DC = deficiency cause, MA = maintenance action, MM = maintenance material, NM = numerical measure, NU = numerical measure unit, QM = categorical quantity measure, SM = categorical severity measure, DT = date.
- The numerical measures and their units follow the format: length, width, height, and area. IN = inch and FT = feet.

Figure 6.4. Examples of the records linked by the proposed data linking method.

6.3.1 Dataset Preparation

A set of ten bridge inspection reports were collected for testing and evaluation. The characteristics of the selected reports are summarized in Table 6.1. These reports are considered representative, because: (1) they are from different regional divisions, representing all five regions in the U.S. (West, Southwest, Midwest, Northeast, and Southeast); (2) they are from different reporting years, ranging from 2006 to 2016; and (3) they are for different types of bridge structures, including steel, masonry, concrete, and timber structures. The information extraction methods, presented in Chapters 4 and 5, were used to extract the records about bridge conditions and maintenance actions from these reports. Errors in the extraction were manually checked and corrected to avoid affecting the data linking evaluation. As a result, a total of 1,743 correct records were included in the dataset. To develop the gold standard annotations for evaluation, the records were independently linked by

four human annotators, who are researchers with background in both civil engineering and machine learning. An initial annotator agreement rate of 78.6% was achieved. Full annotator agreement was then achieved after discussion.

Table 6.1. Characteristics of the selected bridge inspection reports.

Report no.	Region	Structure type	Reporting year	Number of records
1	Midwest	Steel truss arch	2006	409
2	Midwest	Steel multi-girder	2015	152
3	Northeast	Stone masonry arch	2009	97
4	Northeast	Steel truss	2013	451
5	Southeast	Steel cable-stayed bridge	2008	163
6	Southeast	Multi-steel beam with timber deck	2016	93
7	Southwest	Steel girder with concrete abutment	2007	88
8	Southwest	Steel girder with concrete abutment	2008	32
9	West	Deck truss	2011	125
10	West	Double-leaf bascule	2009	133

6.3.2 Evaluation Metrics

The linking results were compared to those in the gold standard, and were evaluated based on example-based precision, recall, and F-1 measure. Using the example-based measures, the data linking performance was calculated for each record in a report, and the overall performance was obtained by calculating the mean performance over all the records in the report. The example-based precision and recall (thereafter called precision and recall for simplification) were calculated using Eqs. (6.5) and (6.6) (Olson and Delen 2008), respectively, where n is the number of records extracted from a report; and, for each record i, true positive (TP) is the number of records correctly-linked to record i, false positive (FP) is the number of records incorrectly-linked to record i, false negative (FN) is the number of records that should but were not linked to record i, $TP + FP$ is the total number of records linked to record i, and $TP + FN$ is the total number of records that should be linked to record i. The example-based F-1 measure, which is the weighted

harmonic mean of precision and recall, was calculated using Eq. (6.7) (Olson and Delen 2008). Average precision, recall, and F-1 measure were calculated as the arithmetic means of the example-based measures over all the reports in the dataset.

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i} \tag{6.5}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{6.6}$$

$$F\text{-}1\ measure = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{6.7}$$

6.3.3 Performance of Term Similarity Scoring Functions

A total of 14 commonly-used term similarity scoring functions (see Section 2.4.2) were evaluated to investigate which one can better assess term similarity. To study the significance levels of their performance differences, a set of Welch's unequal variance t-tests were conducted. For these t-tests, the positional bigram-based function was used as the base case, for two reasons: the pairwise comparisons for all the functions would be unnecessarily too exhaustive to perform, and the chosen function achieved the highest F-1 measure of 92.3%, making it a suitable benchmark. The probability values (p-values) were used to interpret the results of the t-tests: if the p-value is greater than 0.05, there is no significant performance difference between the tested and the base functions; otherwise the difference is significant. Table 6.2 summarizes the experimental results and the p-values.

The results indicate that there is no significant performance difference across the tested functions: all their p-values are greater than 0.05. This could be due to performing stemming before term similarity assessment. The stemming removed the morphological derivations and/or inflections of

the terms, making the functions perform similarly. For example, without stemming, using the exact comparison-based and the Levenshtein edit distance-based functions, the similarity degrees between the terms "cracked" and "cracking" are 0.000 and 0.625, respectively. With stemming, both functions provide the same result (a similarity degree of 1.000), because both terms were converted to the same root form "crack".

Table 6.2. The performance results for the term similarity scoring functions.

Term similarity scoring function [a]	Precision		Recall		F-1 Measure	
	Avg.[b]	P-value [c]	Avg.	P-value	Avg.	P-value
Bag distance	94.1%	0.1477	88.5%	0.7589	91.2%	0.5385
Compression distance	95.3%	0.8465	88.2%	0.6959	91.6%	0.7194
Exact comparison	95.6%	0.9016	87.5%	0.5648	91.4%	0.6513
Jaro distance	93.5%	0.0643	89.7%	0.9471	91.6%	0.6669
Levenshtein edit distance	95.7%	0.8380	89.1%	0.8985	92.3%	0.9578
Longest common substring	95.5%	0.9838	89.2%	0.9239	92.2%	0.9367
Ontology longest common substring	95.2%	0.7792	88.9%	0.8561	91.9%	0.8319
Positional bigram	95.4%	1.0000	89.4%	1.0000	92.3%	1.0000
Sequence matching	95.0%	0.6500	89.2%	0.9176	92.0%	0.8470
Skip-gram	95.8%	0.8098	88.9%	0.8556	92.2%	0.9429
Smith-Waterman edit distance	95.3%	0.8458	89.4%	0.9878	92.3%	0.9513
Syllable alignment distance	95.8%	0.7628	88.4%	0.7433	92.0%	0.8448
Trigram	95.6%	0.9286	88.5%	0.7758	91.9%	0.8261
Winkler distance	92.9%	0.0748	90.0%	0.8628	91.4%	0.6227

[a] For the same function, different parameters were tested and the optimal parameter was selected based on the testing results. For example, for the N-gram-based function, trigram was selected over unigram and bigram. The performance of the function with the selected parameter was reported.
[b] The averages were calculated over the selected ten bridge inspection reports, as per Table 6.1.
[c] The p-values were calculated from the Welch's unequal variance t-tests (using the positional bigram-based function as the base case), and are significant at 0.05 level (2-tailed).

6.3.4 Performance of the Proposed Concept Similarity Scoring Function

The three proposed concept similarity (CS) scoring functions were evaluated to investigate which one can better assess concept similarity. To study the significance levels of the performance differences, a set of Welch's unequal variance t-tests were conducted. Table 6.3 summarizes the experimental results and the p-values.

Two main observations are drawn from these results. First, the similarity of two concepts is better assessed by the similarities of their most-similar terms (as assumed in the term- and relative-position-based functions), rather than the similarities of all their terms (as assumed in the right-position-based function). The former functions achieved an average precision, recall, and F-1 measure of 95.4%, 89.4%, and 92.3%, and 95.5%, 89.5%, and 92.4%, respectively – compared to 100.0%, 67.9%, and 80.9% for the latter one. The right-position-based function decreased the similarity between two concepts when their lengths are different. For example, the function resulted in a low similarity between "strip seal gland" and "gland", because it used all the term pairs and the different concept lengths led to including the dissimilar term pairs ("strip", "gland") and ("seal", "gland") when calculating the concept similarity. As a result, only the true-positive same-length concepts were linked – no false positives were generated; hence the "perfect" precision of 100.0%. Many true-positive different-length concepts were not linked; hence the significant decrease in the recall (by around 21.6%, with p-value = 0.0011). This indicates that the term- and relative-position-based functions are more suitable in assessing concept similarity when the variability in terminology is merely an inconsistency in writing style (e.g., using "longitudinal girder" to refer to "west longitudinal box girder"), such as the case for bridge inspection reports. The right-position-based function would probably be more suitable when variability in terminology is intended, i.e., when different concept lengths indicate different concepts (e.g., concepts at different abstraction levels).

Second, considering the relative positions of terms has a small and insignificant impact on concept similarity assessment. Compared to the term-based function, the relative-position-based function only marginally changed the average precision, recall, and F-1 measure by 0.1%, 0.1%, and 0.2%, respectively, with p-values greater than 0.05. This is mainly because the most-similar terms are

usually placed in similar/same relative positions and their similarities are, thus, not affected by their relative positions. For example, the terms "longitudinal" and "girder" are the most-similar terms between "longitudinal girder" and "west longitudinal box girder", and these terms are in the same relative positions (e.g., both "girder" instances are the last terms).

Table 6.3. The performance results for the proposed concept similarity scoring functions.

CSSF [a]	Precision			Recall			F-1 Measure		
	Avg. [b]	P-value [c]	Delta [d]	Avg.	P-value	Delta	Avg.	P-value	Delta
SF#1	95.4%	–	–	89.4%	–	–	92.3%	–	–
SF#2	95.5%	–	–	89.5%	–	–	92.4%	–	–
SF#3	100.0%	–	–	67.9%	–	–	80.9%	–	–
SF#1 vs. SF#2	–	0.9649	-0.1%	–	0.9842	-0.1%	–	0.9776	-0.1%
SF#1 vs. SF#3	–	0.0003	-4.6%	–	0.0011	+21.5%	–	0.0061	+11.4%
SF#2 vs. SF#3	–	0.0003	-4.5%	–	0.0010	+21.6%	–	0.0058	+11.5%

[a] CSSF = concept similarity scoring function; SF#1 = term-based CSSF; SF#2 = relative-position-based CSSF; SF#3 = right-position-based CSSF.
[b] The averages were calculated over the selected ten bridge inspection reports, as per Table 6.1.
[c] The p-values were calculated from the Welch's unequal variance t-tests, and are significant at 0.05 level (2-tailed).
[d] The delta is the performance difference between the functions in comparison.

6.3.5 Performance of the Proposed Record Similarity Assessment Method

To evaluate the performance of the proposed sequential record similarity assessment method, the classification and clustering results, with and without the use of the method, were compared. Several combinations, as per Table 6.4 (A1 to A6), were evaluated. When not using the method, the normalized total of the concept similarities of two records was used for assessing the record similarity. As shown in Table 6.4, the experimental results indicate that the assessment method was significantly effective, when combined with the improved spectral clustering (without pre-classification) – 6.9% improvement in F-1 measure (A3 vs. A4), with p-value = 0.0044. When combined with the pairwise classification (without clustering), although the method was able to improve the F-1 measure by 12.7% (A1 vs. A2), the improvement was insignificant (p-value = 0.1059) and the improved value was still unsatisfactory (F-1 measure = 59.3%). This is mainly

because the pairwise classification only considered pairwise similarities when linking the concepts/records, without considering the impacts of linking a pair of concepts on the linking of the other concepts (i.e., the intra- and inter-cluster similarities used in the clustering). For example, purely based on the pairwise similarity degree of 1.000 between "truss" and "horizontal strut of truss", the two concepts were incorrectly linked. When considering the following similarities, the "truss" was only linked to the concepts about the same truss, without also being incorrectly linked to the concepts about the strut: the high intra-cluster similarities between "horizontal strut" and "horizontal strut of truss" and between "truss" and "deck truss", and the low inter-cluster similarities between "horizontal strut" and "truss" and between "horizontal strut" and "deck truss". When combined with both, pre-classification and clustering, a similar improvement to that for clustering alone (A3 vs. A4) was shown – 7.2 % improvement in F-1 measure (A5 vs. A6), with p-value = 0.0018, indicating that the proposed sequential record similarity assessment method had a significant impact on clustering only.

Table 6.4. The performance results for the different data linking algorithms.

Data linking algorithm [a]	Precision			Recall			F-1 measure		
	Avg. [b]	P-value [c]	Delta [d]	Avg.	P-value	Delta	Avg.	P-value	Delta
A1	30.5%	–	–	98.4%	–	–	46.6%	–	–
A2	42.8%	–	–	96.5%	–	–	59.3%	–	–
A3	79.2%	–	–	92.3%	–	–	85.2%	–	–
A4	96.2%	–	–	88.3%	–	–	92.1%	–	–
A5	79.0%	–	–	92.3%	–	–	85.1%	–	–
A6	95.4%	–	–	89.4%	–	–	92.3%	–	–
A1 versus A2	–	0.1314	-12.3%	–	0.0497	1.9%	–	0.1059	-12.7%
A3 versus A4	–	0.0000	-17.0%	–	0.2342	4.0%	–	0.0044	-6.9%
A5 versus A6	–	0.0000	-16.4%	–	0.3293	2.9%	–	0.0018	-7.2%
A3 versus A5	–	0.9519	-0.2%	–	0.9982	0.0%	–	0.9550	0.1%
A4 versus A6		0.5363	0.8%		0.7815	-1.1%		0.9069	-0.2%

[a] A1 = pre-classification only (pairwise classification, without clustering); A2 = sequential record similarity assessment + pre-classification; A3 = clustering only (using iterative bi-partitioning spectral clustering, without pre-classification); A4 = sequential record similarity assessment + clustering; A5 = pre-classification + clustering; A6 = sequential record similarity assessment + pre-classification + clustering.

[b] The averages were calculated over the selected ten bridge inspection reports, as per Table 6.1.

[c] The p-values were calculated from the Welch's unequal variance t-tests, and are significant at 0.05 level (2-tailed).

[d] The delta is the performance difference between the algorithms in comparison.

6.3.6 Performance of the Proposed Spectral Clustering-Based Data Linking Method

To evaluate the performance of the improved SC-based data linking method, three main experiments were conducted to: (1) evaluate if the use of unsupervised pre-classification prior to the SC improves the performance of the clustering, (2) select the most suitable threshold value for the partitioning quality assessment function, and (3) evaluate the performance of the iterative bi-partitioning (improved SC).

6.3.6.1 Performance of the Unsupervised Pre-Classification

To evaluate the performance of the unsupervised pre-classification method, the clustering results, with and without the use of pre-classification, were compared. Several combinations, as per Table 6.4 (A3 to A6), were evaluated. The impact of conducting pre-classification, prior to the clustering, on the performance of the data linking was only marginal and insignificant. As shown in Table 6.4, the changes in F-1 measure are smaller than 0.5% (A3 vs. A5 and A4 vs. A6), with p-values much greater than 0.05. This is likely due to two reasons. First, as mentioned above, the pre-classification was formulated as a linear sum assignment optimization task, with the constraint that a concept cannot be linked to itself but has to be linked to another concept. This constraint intended to avoid the situation where all the concepts are only linked to themselves. But, it also resulted in limiting the extent of breaking down the graphs/clusters – the forced links between the concepts prevented further separating the graphs. Second, the sizes of the graphs were not large enough to benefit from size reduction. The pre-classification might show effectiveness in other cases/applications that deal with larger graph sizes (i.e., larger sizes of records per report) (Liu et al. 2013; Chen and Cai 2015).

6.3.6.2 Threshold Analysis

To select the optimal threshold value for the partitioning quality assessment function, a total of 21 experiments were conducted, with threshold values ranging from 0 to 1 and a step size of 0.05. The experimental results, as shown in Figure 6.5, indicate that (1) when the threshold value increased from 0.00 to 0.05, the F-1 measure increased; (2) when it was set between 0.05 and 0.15, the F-1 measure did not change much; and (3) starting at a value of 0.15, as the value increased, the F-1 measure showed a decreasing trend. The optimal threshold value is, thus, within the range of 0.05 and 0.15. As the threshold value becomes smaller than the optimum, accepting a partitioning becomes easier, which results in over-partitioned clusters that contain less false positives and much more false negatives. This leads to slowly-increased precision and quickly-decreased recall, which decreases the F-1 measure. As the threshold value becomes greater than the optimum, accepting a partitioning becomes more difficult, which results in under-partitioned clusters that contain more true positives and much more false positives. This leads to quickly-decreased precision and slowly-increased recall, which decreases the F-1 measure. For the following experiments, a threshold value of 0.10 was used.

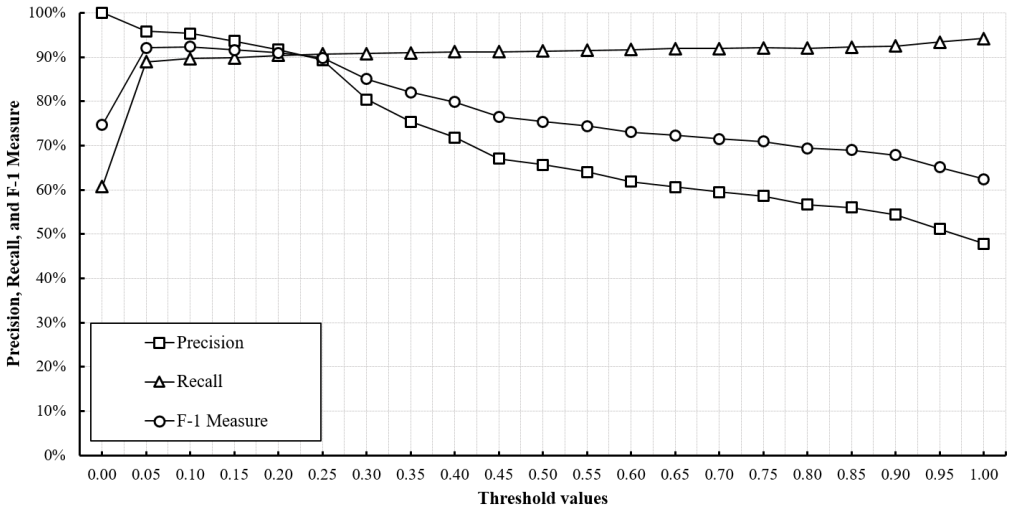


Figure 6.5. Performance of different threshold values.

6.3.6.3 Performance of the Iterative Bi-Partitioning

To evaluate the performance of the iterative bi-partitioning method, it was compared to the elbow method. Using the elbow method, the optimal number of target clusters is manually selected based on the R-squared (RS) index, which is the ratio of the intra-cluster variance to the total variance of a dataset (Gan et al. 2007). As shown in Table 6.5, the iterative bi-partitioning method was effective in correctly identifying the numbers of target clusters and in clustering the records. Compared to those identified by the elbow method, the numbers identified by the bi-partitioning method were much closer to the true numbers in the gold standard. In terms of linking performance, the bi-partitioning method significantly improved the precision by 46.2%, with p-value = 0.0000.

The iterative bi-partitioning method outperformed the elbow method because of two main reasons. First, it used a "one-vs-all" (OVA)-like clustering approach, where binary clustering decisions for partitioning the records into two clusters were iteratively made; while, the elbow method used a "one-vs-one" (OVO)-like approach, where a "multi-class" clustering decision for partitioning the records into a large number of clusters were made at once. Making a sequence of binary decisions is less complex than making a "multi-class" decision at the same time. The clustering approaches also affected the performance of the k-means clustering method, which is used in the original NJW normalized SC method to cluster the leading eigenvectors of the Laplacian matrix. The k-means method only needed to consider two (the number of target clusters) seeds using the OVA approach, but it needed to consider a large number of seeds using the OVO approach. It has been shown that the k-means method becomes unstable and less accurate when dealing with a large number of seeds (Boukhdhir et al. 2015). Second, the partitioning quality assessment function used in the bi-partitioning method considered both the inter- and intra-cluster similarities to identify the optimal numbers and cluster the records. Compared to the RS index that only considered the intra-cluster

variance, this function was thus able to better capture the dynamic nature of the clustering/linking – the impact of linking a set of records on the linking of the other records. In addition, unlike the elbow method, which requires human interpretation to identify the "elbow" points, the assessment function only uses an experimentally-determined threshold value, which avoided the subjectivity in selecting the number.

Table 6.5. The performance results for the iterative bi-partitioning method.

Report no. [a]	Elbow method			Iterative bi-partitioning method			Number of clusters in the gold standard
	Precision	Recall	Number of clusters identified	Precision	Recall	Number of clusters identified	
1	50.5%	76.9%	111	97.9%	76.3%	217	172
2	74.5%	77.9%	51	93.5%	86.7%	59	54
3	55.4%	86.0%	37	96.6%	90.9%	60	57
4	39.7%	77.3%	116	94.6%	73.5%	299	249
5	15.9%	99.4%	13	95.3%	89.8%	111	106
6	55.5%	95.2%	33	98.6%	97.8%	58	57
7	37.5%	95.9%	25	97.7%	97.0%	65	66
8	44.8%	96.9%	12	100.0%	96.9%	29	27
9	61.5%	83.5%	54	93.7%	87.2%	86	74
10	64.9%	79.8%	76	93.7%	87.1%	95	90
Average	50.0%	86.9%	53	96.2%	88.3%	108	95
P-value	0.0000 [b]	0.5172 [b]	0.1021 [c]	–	–	0.7141 [c]	–

[a] The report numbering follows that defined in Table 6.1.
[b] The p-values for comparing the precision (or the recall) of the iterative bi-partitioning method to that of the elbow method.
[c] The p-values for comparing the number of clusters identified using the iterative bi-partitioning method (or the elbow method) to that in the gold standard.

6.3.7 Overall Performance of the Proposed Data Linking Method and Error Analysis

The overall performance results for the proposed data linking method are shown in Table 6.6. The results are based on using the positional bigram-based term similarity scoring function and the term-based concept similarity scoring function. On average, the algorithm achieved a precision, recall, and F-1 measure of 96.2%, 88.3%, and 92.1%, respectively.

Table 6.6. Performance results for the proposed data linking algorithm.

Report no. [a]	Precision	Recall	F-1 measure
1	97.9%	76.3%	85.7%
2	93.5%	86.7%	90.0%
3	96.6%	90.9%	93.6%
4	94.6%	73.5%	82.7%
5	95.3%	89.8%	92.5%
6	98.6%	97.8%	98.2%
7	97.7%	97.0%	97.4%
8	100.0%	96.9%	98.4%
9	93.7%	87.2%	90.3%
10	93.7%	87.1%	90.3%
Average	96.2%	88.3%	92.1%

[a] The report numbering follows that defined in Table 6.1.

Two main sources of errors that caused false negatives were identified. First, the concept similarity scoring function was limited when the similarities of the most-similar terms cannot sufficiently capture the similarities between the concepts. For example, a concept similarity of 1.000 is expected for the concepts “upper bracing” and “top bracing”, because they are used to refer to the same bracing entity. But, the function assessed the concept similarity as 0.500 due to the similarity of 0.000 for the most-similar terms (“upper”, “top”), making the concepts incorrectly unlinked. Second, the partitioning quality assessment function cannot successfully deal with ambiguous assessment cases, in which the inter-cluster similarity between two child clusters is exactly 0.500 (a cut-off between similar and dissimilar) and the intra-cluster similarity of each child is 1.000. In such cases, the function tended to over-partition parent clusters, which caused false negatives. For example, the inter-cluster similarity between the concepts about “rubber trough” and the concepts about “drain trough” is 0.500 and the intra-cluster similarity is 1.000. These similarities led to a PQA score that is greater than the threshold value, making the two set of concepts incorrectly unlinked.

One main source of errors that contributed to the false positives was identified. The concept similarity scoring function was limited in some cases when a concept is composed of a noun phrase and a prepositional phrase. For example, the scoring function incorrectly assigned a similarity of 1.000 to “upper strut of tower” and “tower”, making these two concepts incorrectly linked.

CHAPTER 7 – HYBRID DATA FUSION

This chapter presents the proposed data fusion method for fusing the linked data records (extracted as per Research Tasks #3 and #4, and linked as per Research Task #5) into a unified representation. The method development and evaluation (Research Task #6) are presented in this chapter. The integration method for integrating the fused data with the other types of structured data (i.e., NBI and NBE, data as well as traffic and weather data) is presented in Chapter 8, Section 8.2.2.

7.1 Comparison to the State of the Art

Fusing the data extracted from textual bridge inspection reports requires two tasks. First, concept names that refer to the same entity, but vary in terms of surface forms and abstraction levels, need to be fused into canonical identifier names. This is different from concept mapping (e.g., Zhang and El-Gohary 2016; Le and Jeong 2017), which focuses on classifying the types of relationships between concept names and mapping equivalent concept names together. Rather, this is a concept naming problem – representing the concept names using canonical identifier names that balance the abstraction and detailedness, so that they are not too frequent or too rare (in a collection of inspection reports) to the extent of causing the loss of distinctive feature patterns or undermining the generalizability. Fusing concept names was thus defined, in this research, as a named entity normalization task: the multiple concept names that are used in a single report to refer to the same entity are normalized into a canonical identifier with balanced abstraction and detailedness, and the identifiers from different reports are subsequently fused if they are the same. Second, the numerical deficiency measures of the multiple instances, which are of the same type of deficiency but are at different locations of a bridge element, need to be fused into a single representative representation. Unlike data in multi-sensor data fusion applications (e.g., Jiang et al. 2016; Zhang et al. 2017), which are mainly characterized as being conflicting, imprecise, and/or multi-modal

(Khaleghi et al. 2013), each of the deficiency measures is partially describing the overall condition of the deficiency and these data are, thus, complementary. Fusing deficiency measures was thus defined, in this research, as a numerical data fusion task: the measures of the multiple deficiency instances, from one report, are fused into a single representation that is representative of all the original measures.

7.2 Data Fusion Method Development

7.2.1 Proposed Data Fusion Method

A hybrid data fusion method is proposed. At the cornerstone of the method are two proposed algorithms for fusing concept names and numerical data, respectively: an unsupervised named entity normalization algorithm and an entropy-based numerical data fusion algorithm. As depicted in Figure 7.1, the input of the proposed method is data records that are extracted from different bridge inspection reports and are linked if they refer to the same entity and come from the same report. The method includes two main steps for fusing these records. First, for a set of linked records, the concept names are fused into a canonical name with balanced abstraction and detailedness using the proposed normalization algorithm, resulting in a set of partially-fused records. The canonical names from all the partially-fused records are fused if they are the same, and the fused names are used as features in the unified representation of the reports. Second, for a set of partially-fused records, the numerical deficiency measures are fused into a single representative interval-based representation using the proposed fusion algorithm, resulting in a fully-fused record. The fused data from all the fully-fused records are used as values of their corresponding features/names and inspection reports in the unified representation.

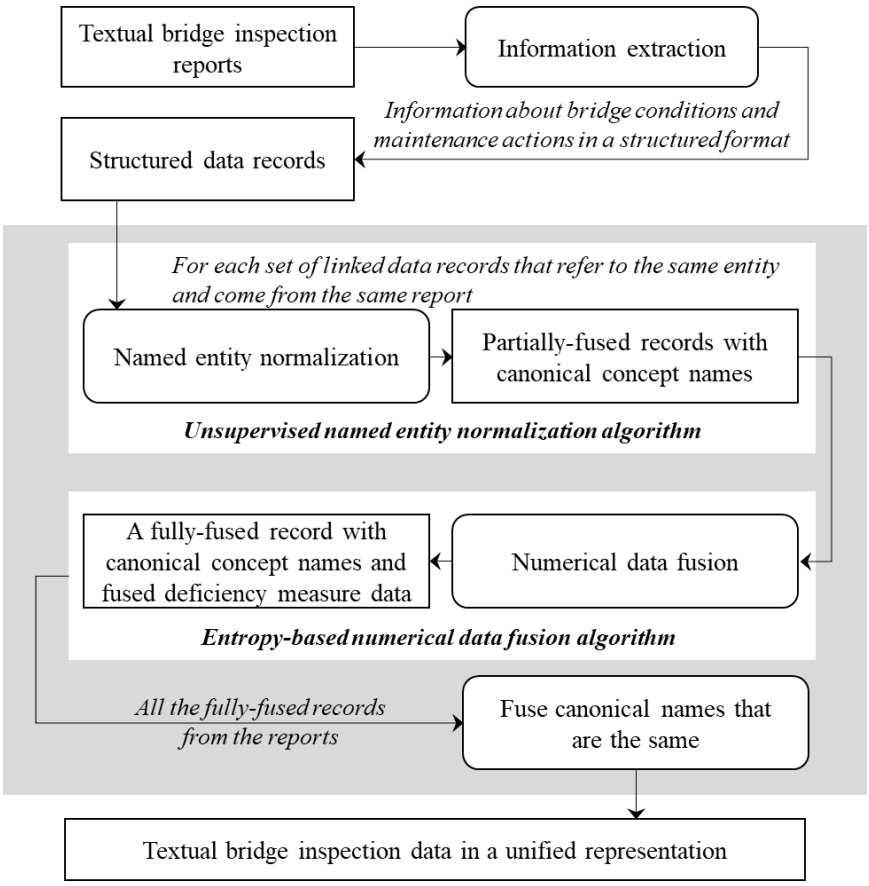


Figure 7.1. Overview of the proposed hybrid data fusion method.

7.2.1.1 Unsupervised Named Entity Normalization

A new unsupervised named entity normalization algorithm is proposed. It fuses concept names that refer to the same entity, but vary in terms of both surface forms and abstraction levels, into a canonical identifier concept name that balances the abstraction and detailedness. The proposed algorithm includes three primary components: identifier concept name generation, ranking, and selection. Figure 7.2 provides an overview of the proposed normalization algorithm.

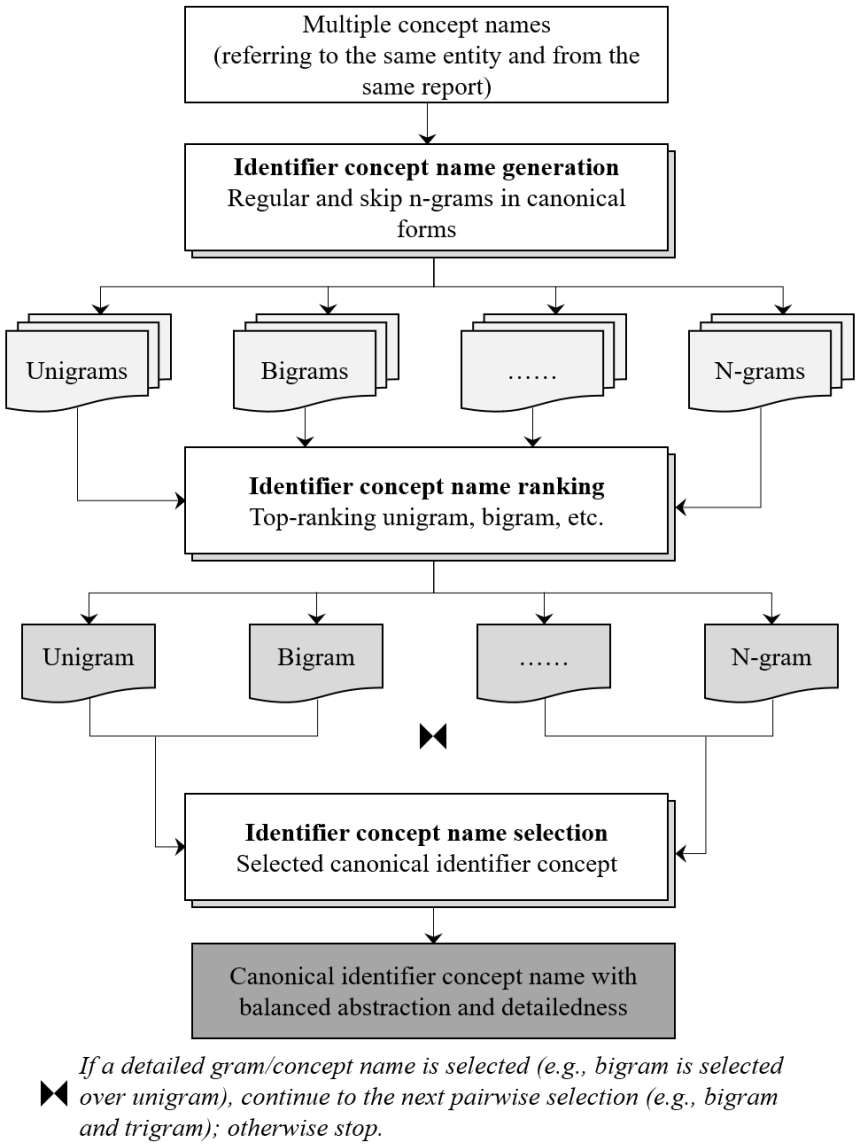


Figure 7.2. Proposed unsupervised named entity normalization algorithm.

7.2.1.1.1 *Identifier Concept Name Generation*

Identifier concept name generation aims to generate all candidate identifier concept names – in their canonical forms and at different abstraction levels – that a set of original concept names could have. The generation includes two steps: morphological analysis and n-gram generation. Morphological analysis aims to analyze how a term is formed based on morphological derivation and inflection, and to map the term into its canonical form. It is used to account for the surface-

form variations. For example, for “bridge railing” and “bridge rail”, morphological analysis removed the suffix of “railing” and mapped the first name to its canonical form “bridge rail”, resulting in a normalized surface form of the two. N-gram generation aims to generate candidate identifier names which are at different abstraction levels, so that an identifier that balances the abstraction and detailedness can be subsequently selected. It is used to capture the abstraction-detailedness variations. Two types of candidate names are generated from the original names (in canonical forms) using an n-gram language model: regular and skip n-grams. Regular n-grams are the concept names (e.g., unigram, bigram, and trigram concept names) that have constituent terms following the same consecutive sequence as they appear in an original concept name. Skip n-grams are similar to regular n-grams, but their terms are not consecutive in the original name. For example, “asphalt deck” and “asphalt wearing” are the regular and skip bigrams of the concept name “asphalt deck wearing surface”, respectively.

7.2.1.1.2 *Identifier Concept Name Ranking*

Identifier concept name ranking aims to rank the generated candidate identifier concept names. The ranking is, separately, conducted at each abstraction level. For example, bigram names (both regular and skip) are ranked separately – not together with the other types of names (e.g., unigram and trigram names) – to avoid the mixing of concept name distributions, which would negatively affect the ranking. A new concept ranking function is, thus, proposed to rank the candidate identifiers. As shown in Eq. (7.1), the proposed function considers the corpus statistic score (*CSS*), term-position score (*TPS*), and term-sequence score (*TSS*) of a candidate identifier concept name (*CICN*) to calculate its ranking score.

$$Ranking\ score(CICN) = CSS(CICN) \times TPS(CICN) \times TSS(CICN) \quad (7.1)$$

The corpus statistic score is used to rank the candidate concept names based on how frequent or rare they are in a collection of bridge inspection reports. To calculate the scores, two alternative corpus statistic measures, term frequency (TF) and inverse document frequency (IDF), were selected. TF captures the frequency rate of a concept name in all the sets of candidate names, where each set contains the candidate names that refer to the same entity and come from a single report in the collection. It prefers the concept names that are frequent. IDF captures the frequency rate of a concept name across all the sets, i.e., how many sets in the collection contain a specific name. It prefers the concept names that are less frequent across the sets and are thus rare. Two variations of the measures, TF-IDF and Okapi BM25, were also selected because, theoretically, they can balance both types of preferences. The performances of these four measures were tested (see Section 7.3.1.1).

The term-position and term-sequence scores are used to rank the candidate concept names based on how meaningful they are, because it is desirable for names that are meaningful to be ranked high. They are calculated based on the lexical patterns (i.e., lexical position and sequence) of terms in their original concept names. The term-position scores are calculated based on the following lexical-position hypothesis: the contribution of a term's meaning to the entire meaning of a concept name decreases from right to left; the most right-hand side term contributes the most (Zhang and El-Gohary 2016). Thus, a candidate concept name that is mostly composed of terms from the right-hand side of an original name has a higher score than the name that is mostly composed of terms from the left-hand side. The term-position score is calculated using Eq. (7.2), where *CICN* is a candidate identifier concept name, *OCN* is an original concept name in a set of original names *OCNs*, *N* is the number of names in the set, *T* is a term of *CICN*, *M* is the number of terms in *CICN*, $Index_{OCN}(T)$ is the index of *T* in *OCN*, and $|OCN|$ is the length of an original concept name.

$$Term\text{-}position\ score\ (CICN) = 1.0 + \frac{1}{N}\sum_{\{OCN \in OCNs\}}\frac{1}{M}\sum_{\{T \in CICN\}}\frac{Index_{OCN}(T)}{|OCN|} \quad (7.2)$$

The term-sequence scores are calculated based on the following lexical-sequence hypothesis: a candidate concept name with terms following the same consecutive sequence as they appear in its original name has a higher score. This hypothesis was made because using skip n-grams, although provides more candidate concept names with various term combinations, generates some names that have terms that do not follow the same consecutive sequence and are, thus, generally less meaningful. For example, the terms of the skip bigram "asphalt wearing" do not follow the same consecutive sequence as they appear in the original concept name "asphalt deck wearing surface" and are, thus, less meaningful. The term-sequence score is calculated using Eq. (7.3), where $I_{\{1,0\}} = 1.0$ if a candidate concept name has terms following the same consecutive sequence as they appear in an original name; otherwise $I_{\{1,0\}} = 0$. The other notations follow those defined in Eq. (7.2).

$$Term\text{-}sequence\ score(CICN) = 1.0 + \frac{1}{N}\sum_{\{OCN \in OCNs\}} I_{\{1,0\}} \quad (7.3)$$

7.2.1.1.3 *Identifier Concept Name Selection*

Identifier concept name selection aims to select a final canonical identifier concept name from the top-ranking identifier names (one top-ranking name for each abstraction level). The selection is conducted hierarchically (i.e., in a top-down fashion), so as to select an identifier with balanced abstraction and detailedness. For example, for a pair of top-ranking identifiers in the adjacent abstraction levels (e.g., unigram and bigram names), if the detailed name fails to meet any of the if statements in a proposed selection rule, the abstract name is selected as the final identifier;

otherwise, the selection continues to the next pair (e.g., bigram and trigram names), until the abstract name in the pair is selected or no detailed name is available.

The proposed concept selection rule, which considers both the corpus statistics and the lexical patterns of the concept names, includes three cascading if statements:

- *if the ranking score of the detailed concept name added by an adjustment factor alpha is greater than the ranking score of the abstract concept name*; the adjustment factor alpha is used to balance the abstraction and detailedness of the identifier concept names. A large value of alpha favors detailed names, and a small value favors abstract names.
- *if the word-association score of the detailed concept name is greater than a threshold value beta*; the word-association score measures the degree to which two terms are related, using corpus statistics (e.g., co-occurrence rates of the terms in a collection of inspection reports). It is used to make sure that the detailed concept names are lexical atoms (semantically-coherent phrases, e.g., "map crack") rather than random combinations of terms. The normalized Google distance (Cilibras and Vitany 2007) was selected to calculate the word-association scores, because it is less negatively affected by extremely-frequent terms, which make lexical atoms have low scores as random combinations. The threshold value beta is used to further balance the abstraction and detailedness. A large value of beta makes it stringent for detailed identifier names to be selected and, thus, favors abstract names. A small value makes it easier for detailed names to be selected and, thus, favors such names.
- *if the part-of-speech (POS) pattern of the detailed concept shows a noun-phrase pattern*; the POS patterns (i.e., lexical class patterns of terms) are used to filter out concept names that are not noun phrases, because a noun phrase is the most frequently-occurring phrase type and is

commonly used for naming concepts. Two noun-phrase patterns were used: “noun + noun” and “adjective + noun”, where “noun” could be a noun or a noun phrase.

As noted, two hyperparameters, alpha and beta, are used in the rule to balance the abstraction and detailedness of the candidate identifier concept names. In order to find the optimized values for them, a total of 10,000 value combinations (with the values of alpha and beta ranging from 0 to 1 with step size of 0.01, respectively) were tested. The combination with alpha = 0.38 and beta = 0.81 was empirically selected based on the testing results and was used for the experiments conducted in this research.

7.2.1.2 Entropy-Based Numerical Data Fusion

A new entropy-based numerical data fusion algorithm is proposed to fuse multiple numerical data into a single representative interval-based representation. The proposed algorithm includes four primary components: interval determination, degree tuple quantification, degree tuple fusion, and interval-based data representation. Figure 7.3 shows the proposed fusion algorithm.

7.2.1.2.1 *Interval Determination*

Interval determination aims to determine the number of intervals and the size of the interval for representing the fused data. Intervals are used to represent the fused data, in order to account for the uncertainty in data and to avoid the exaggerated impact of minor fluctuations in continuous data on the machine learning models. The proportional k-interval discretization method (Yang and Webb 2001) is used to define the intervals. This method was selected because it can use data to balance the trade-off relationship between the number of intervals and the interval size. A large number is preferred to capture more distinctive data instances for avoiding underfitting; and, at the same time, a larger size is preferred to retain more data instances within an interval for avoiding

overfitting. However, as the number increases, the size decreases. To balance this, the discretization method gives equal weight to them. The number of the intervals is defined as $\sqrt{K}$ and the size of each interval is defined based on the minimum and the maximum of the $\sqrt{K}$ unique data instances in the interval, where K is the number of unique data instances in a dataset (e.g., unique deficiency measures of the same type of deficiency in a collection of inspection reports).

Algorithm: Entropy-based numerical data fusion algorithm

1: **Input** All the unique numerical data instances in the dataset
2: **Execute** Interval formalization (i.e., defining the number of intervals and the interval size using the proportional k-interval data discretization)
3: **Output** A set of intervals: $\{I_i\}, i = 1, \dots, M$
4: **Input** A set of numerical data instances to be fused
5: **Execute** Degree tuple quantification, as per Eq. (4)
6: **Output** Degree tuple matrix (*DTM*)
7: $$DTM = \begin{bmatrix} DT_{1,1} & \dots & DT_{1,N} \\ DT_{2,1} & \dots & DT_{2,N} \\ \vdots & \ddots & \vdots \\ DT_{M,1} & \dots & DT_{M,N} \end{bmatrix}$$
8: where DT is a degree tuple calculated as per Eq. (4), N is the number data instances, and M is the number of intervals.
9: **Execute** Degree tuple fusion, as per Eq. (5)
10: **Output** Fused degree tuple vector (*FDTV*)
11: $$FDTV = \begin{bmatrix} FDT_1 \\ FDT_2 \\ \vdots \\ FDT_M \end{bmatrix}$$
12: where *FDT* is a fused degree tuple calculated as per Eq. (5).
13: **Execute** Interval-based data representation
14: **For** each fused degree tuple
15: **Computed** its Euclidean between the ideal degree tuple [1, 0, 0]
16: **Output** the interval corresponding to the tuple with the smallest distance
17: **Output** Unified representation:
the count of data instances, the single representative interval of the data

Figure 7.3. Proposed entropy-based numerical data fusion algorithm

7.2.1.2.2 *Degree Tuple Quantification*

Degree tuple quantification aims to quantify the values contained in a degree tuple: membership, non-membership, and indeterminacy degree values. Membership and non-membership degrees are the extent of a data instance belonging and not belonging to an interval, respectively.

Indeterminacy degree is the extent of hesitancy in claiming that the instance belongs or does not belong to the interval. The normal cloud model (Li et al. 2009) is used to quantify these values, because it can capture the uncertainty in the membership and non-membership to allow for the modeling of indeterminacy. The normal cloud model, which is based on the Gauss membership function and normal distribution, is a generalized normal distribution for quantifying the membership degree of a data instance belonging to an interval as a value between 0 and 1 (Li et al. 2009). The model assumes that the standard deviation of the Gauss membership function is not a fixed number, but a random number following a normal distribution. Because of the randomness in drawing the standard deviation, for an interval, the Gauss function maps a data instance to many membership degree values (i.e., one-to-many mapping). Based on this mapping property, a new equation is proposed to quantify the degree tuple, as per Eq. (7.4), where x is a data instance, I is an interval, $u_I(x)$ is a membership degree value mapped from the Gauss function, and MDV, NDV, and IDV are the membership, non-membership, and indeterminacy degree values of x to I, respectively.

$$\begin{bmatrix} MDV_I(x) \\ NDV_I(x) \\ IDV_I(x) \end{bmatrix} = \begin{bmatrix} min(\{u_I(x)\}) \\ 1 - max(\{u_I(x)\}) \\ max(\{u_I(x)\}) - min(\{u_I(x)\}) \end{bmatrix} \tag{7.4}$$

7.2.1.2.3 *Degree Tuple Fusion*

Degree tuple fusion aims to fuse the quantified degree tuples of an interval into a single tuple. An information entropy-based fusion function is proposed to conduct the fusion. Information entropy is the average rate at which a stochastic process generates information (Shannon 1948); intuitively, it measures the amount of information in a random variable, where information entropy equal to zero indicates that the variable always generates the same information (Mehri and Darooneh 2011).

Considering an interval as a variable that generates data instances, if it always generates the same particular instance (i.e., the membership degree value of this instance to the interval is 1) and cannot generate other instances (i.e., the membership degree values are 0), the information entropy of this interval is zero. Conversely, if the interval generates all the instances (i.e., the membership values of these instances to the interval are within 0 and 1), its information entropy is greater than zero. In the first case, the interval can only represent the particular instance and, thus, is less representative of all the complementary data instances that collectively describe a target (e.g., the overall condition of a deficiency). As a result, its fused membership degree value should be downweighed and the other two values should be upweighted. In the second case, the interval can represent all the instances and, thus, is more representative. As a result, its fused membership degree value should be upweighted and the other two values should be downweighed. Based on the aforementioned analysis, the proposed information entropy-based function fuses the degree tuples of an interval as per Eq. (7.5), where W_I is the weight of the interval calculated using Eq. (7.6), i.e., the information entropy of the interval divided by the sum of information entropies of all the generated intervals. In Eqs. (7.5) to (7.6), $FMDV$, $FNDV$, and $FIDV$ are the fused membership, non-membership, and indeterminacy degree values of interval *I*, respectively; N is the number of instances in the set of numerical data X; and M is the total number of intervals generated from the data discretization. The other notations follow those defined in Eq. (7.4).

$$\begin{bmatrix} FMDV_I(X) \\ FNDV_I(X) \\ FIDV_I(X) \end{bmatrix} = \begin{bmatrix} \frac{W_I}{N} \sum_{\{x \in X\}} MDV_I(x) \\ \frac{1 - W_I}{N} \sum_{\{x \in X\}} NDV_I(x) \\ \frac{1 - W_I}{N} \sum_{\{x \in X\}} IDV_I(x) \end{bmatrix} \tag{7.5}$$

$$W_I = \frac{\sum_{i=1}^{N} MDV_I(x) \times log_2 MDV_I(x)}{\sum_{j=1}^{M}\sum_{i=1}^{N} MDV_{I_j}(x) \times log_2 MDV_{I_j}(x)} \quad (7.6)$$

7.2.1.2.4 *Interval-Based Data Representation*

Interval-based data representation aims to select an interval from all the possible intervals (defined in Section 7.2.1.2.1) for representing the multiple numerical data instances. The selection is conducted based on the Euclidean distance between the fused degree tuple and the ideal degree tuple [1, 0, 0]. An ideal degree tuple has a fused membership degree value 1 and the other two degree values 0. This entails that the ideal interval corresponding to the ideal tuple can fully represent the multiple data instances that are complementary. Thus, an interval is selected if its fused degree tuple is the closest to the ideal tuple. As a result of the numerical data fusion process, the set of multiple numerical data instances are represented in a united way as: the count of the data instances, the single representative interval of the instances.

7.2.2 Implementation of the Proposed Method

7.2.2.1 Implementation for Method Verification

The verification aimed to evaluate the correctness of the proposed hybrid data fusion method. The verification included two main steps: dataset preparation and verification experiments. Two types of experiments were conducted to verify the two algorithms respectively: named entity normalization experiments and numerical data fusion experiments.

7.2.2.1.1 *Dataset Preparation*

A dataset, which includes ten bridge inspection reports, was created. The information about these reports is summarized in Table 6.1. The information extraction methods (developed as Research Tasks #3 and #4, in Chapters 4 and 5, respectively) were used to extract information about bridge

conditions and maintenance actions from these reports and to represent the extracted information in a structured format. The data linking method (developed as per Research Task #5, in Chapter 6) was used to link the extracted data records that refer to the same entity and come from the same report. The linked records formed the dataset for the normalization and fusion experiments.

7.2.2.1.2 *Named Entity Normalization Experiments*

The experiments aimed to implement the proposed normalization algorithm to evaluate its accuracy, by comparing the algorithm-generated identifier concept names to the gold-standard identifiers. The algorithm was implemented in a Python program (Python version 2.7). The natural language toolkit Porter stemmer and "ngrams" function (Bird et al. 2009) were used for the morphological analysis and the n-gram generation, respectively. The Stanford POS tagger (Toutanova et al. 2003) was used for analyzing the POS patterns of the concept names. The gold standard was prepared by three human annotators, who are researchers with background in civil engineering, natural language processing, and machine learning. Full inter-annotator agreement was achieved after discussion. Accuracy, which is the number of correct identifier concept names out of the total number of identifier concept names, was calculated using Eq. (7.7).

$$Accuracy = \frac{Number\ of\ correct\ identifier\ concept\ names}{Total\ number\ of\ identifier\ concept\ names} \tag{7.7}$$

7.2.2.1.3 *Numerical Data Fusion Experiments*

The fusion experiments aimed to implement the proposed fusion algorithm to evaluate its stability in Monte Carlo simulations. Two main factors could affect the stability of the algorithm: the uncertainty in the data and the randomness in drawing the standard deviation of the Gauss function. Thus, two main groups of simulations were conducted: (1) simulations with data sampled from normal distributions, where each sampled instance has an uncertainty level (i.e., the standard

deviation of the normal distribution) ranging from 0.5 to 10 with a step size of 0.5, and (2) simulations with the times of randomly drawing the standard deviation of the Gauss function ranging from 100 to 2,000 with a step size of 100. The number of iterations for each simulation run was set to 10,000. The algorithm and simulations were implemented in a Python program (Python version 2.7). Information entropy was used to evaluate the stability of the algorithm. It is equal to zero if the algorithm can stably fuse the same set of data instances into the same interval in a simulation run; otherwise, it increases. As a verification metric, it was calculated using Eq. (7.8) (Pathria and Beale 2011), where M is the number of intervals, N_i is the times of the i^{th} interval being selected to represent the same set of data instances , and N is the number of iterations in a simulation run (i.e., $N = 10{,}000$).

$$Information\ entropy = -\sum_{i=1}^{M} \frac{N_i}{N} \times log_2 \frac{N_i}{N} \tag{7.8}$$

7.2.2.2 Implementation for Method Validation

The validation aimed to evaluate the performance of the proposed hybrid data fusion method in supporting its intended use, fusing data extracted from bridge inspection reports for supporting enhanced bridge deterioration prediction.

7.2.2.2.1 *Dataset Preparation*

The NBI data and the textual bridge inspection reports of 1,300 bridges, which are located in the state of Washington, were collected. The NBI data were collected from the Federal Highway Administration (FHWA 2019). The NBI data have a total of 134 features, including features about bridge location, geometric characteristics (e.g., bridge length, deck width, and number of spans, etc.), structural characteristics (e.g., functional classification, design load, wearing surface type, etc.), construction characteristics (e.g., year built and type of construction), conditions (i.e.,

condition ratings), etc. The condition ratings use a 0-9 scale to describe the condition of a primary bridge component (a deck, a superstructure, or a substructure), with 0 and 9 representing "failed condition" and "excellent condition", respectively. A description of the meaning of each condition rating category is presented in Table 7.1. For more details about the NBI data, the readers are referred to FHWA (1995). The bridge inspection reports were collected from the Washington Department of Transportation. Same as the dataset preparation conducted for method verification, information extraction and data linking were conducted to process the reports. The linked records were then fused, as per Figure 7.1, thereby forming the unified representation of the data extracted from the reports. Using the collected data, seven datasets were created. Table 7.2 summarizes the details of these datasets. In each dataset, the data were split into a training dataset and a testing dataset. The training dataset contains the data from 2013, and the condition ratings of the decks, superstructures, and substructures of the bridges from 2015. The testing dataset contains the data and the ratings from 2015 and 2017, respectively.

Table 7.1. Description of the condition rating categories in National Bridge Inventory data [1].

Condition rating category	Description
9	Excellent condition
8	Very good condition
7	Good
6	Satisfactory condition
5	Fair condition
4	Poor condition
3	Serious condition
2	Critical condition
1	"Imminent" failure condition
0	Failed condition

[1] The condition rating categories and its corresponding description are defined by FHWA (1995).

Table 7.2. Summary of the created datasets.

Dataset	Data [1]	Purpose [2]
#1	The national bridge inventory (NBI) data	Used to develop baseline prediction models to evaluate if further learning from bridge inspection reports is able to enhance the performance of bridge deterioration prediction.
#2	The NBI data + the unfused data extracted from the bridge inspection reports	Used to develop baseline prediction models to evaluate if the fusion of the data extracted from reports is necessary for enhance the prediction performance.
#3	The NBI data + the report data fused by the proposed hybrid data fusion method	Used to develop prediction models to evaluate the performance of the proposed method in fusing the data extracted from the reports for supporting enhanced bridge deterioration prediction.
#4	The NBI data + the fused report data (where the deficiency measures were fused by taking the maximum of the measures, i.e., using the worst deterioration case)	Used to develop baseline prediction models to evaluate if the deficiency measures fused by the proposed method can better support the prediction, compared to the measures fused by taking the maximum.
#5	The NBI data + the fused report data [where the deficiency measures were fused by using one of the central tendency measures (arithmetic mean, Bonferroni mean, geometric mean, harmonic mean, Heroin mean, power mean, median, mode)]	Used to develop baseline prediction models to evaluate if the deficiency measures fused by the proposed method can better support the prediction, compared to the measures fused by using the mean (or total, i.e., the data were represented as the number of measures instances + the mean of the measures).
#6	The NBI data + the fused report data [where the deficiency measures were fused by using one of the variation measures (range, mean absolute difference, coefficient of variation, standard deviation, variance)]	Used to develop baseline prediction models to evaluate if the deficiency measures fused by the proposed method can better support the prediction, compared to the measures fused by using the variation.
#7	The NBI data + the fused report data [where the deficiency measures were fused by using one of the central tendency measures] + the fused report data [where the deficiency measures were fused by using one of the variation measures]	Used to develop baseline prediction models to evaluate if the deficiency measures fused by the proposed method can better support the prediction, compared to the measures fused by using the combinations of the central tendency and the variation measures (e.g., the measures were represented as the number of the measure instances, the arithmetic mean of the measures, the variance of the measures).

[1] The concept names in the textual bridge inspection reports in datasets #4 to #7 were fused by the proposed unsupervised named entity normalization algorithm.
[2] Bridge deterioration prediction means the predictions of future condition ratings of decks, superstructures, and substructures.

7.2.2.2.2 *Validation Experiments*

The validation experiments aimed to develop machine learning models for predicting the future condition ratings of decks, superstructures, and substructures. The decision tree algorithm was

selected for developing the models, because it can directly handle both categorical and numerical features, without the need for one-hot encoding. One-hot encoding transforms categorical features into numerical features using dummy variables. Such variables increase the dimensionality and the sparsity of the feature space, which would negatively affect the validation. Seven main types of the prediction models were developed, with each type trained and tested using the data in one of the datasets, as per Table 7.2. Average accuracy was selected as the validation metric. Average accuracy is the average of the ratio of the number of correctly-predicted condition ratings to the total number of ratings per condition rating category. It was calculated using Eq. (7.9), where N is the number of condition rating categories, CRs are condition ratings, and CRC is a condition rating category.

$$Average\ accuracy = \frac{1}{N}\sum_{i=1}^{N} \frac{Number\ of\ correctly\text{-}predicted\ CRs\ in\ the\ i^{th}\ CRC}{Total\ number\ of\ CRs\ in\ the\ \ i^{th}\ CRC} \quad (7.9)$$

7.3 Data Fusion Method Evaluation

7.3.1 Performance Results of Method Verification

7.3.1.1 Performance of the Proposed Named Entity Normalization Algorithm

Table 7.3 summarizes the performance results of the proposed normalization algorithm. The results show that the algorithm performed well: it achieved an average accuracy of 94.4%. Two important observations were also drawn from the results.

Table 7.3. Performance results for the proposed named entity normalization method.

Ranking function [1]	Part-of-speech (POS) pattern [2]	Accuracy for each concept name type [3]				
		ET (%)	DY (%)	DC (%)	MA (%)	MM (%)
CSS	–	69.4	86.3	93.7	99.1	100.0
	"Adj. + noun"	64.6	54.9	61.7	99.1	79.3
	"Noun + noun"	73.3	80.5	89.9	99.1	100.0
	"Adj. + noun" & "noun + noun"	69.9	83.2	93.7	99.1	88.9
CSS x TPS	–	71.3	86.4	93.7	99.1	100.0
	Adj. + noun	64.7	54.9	61.7	99.1	79.3
	Noun + noun	75.1	80.6	89.9	99.1	100.0
	Adj. + noun & noun + noun	71.8	83.2	93.7	99.1	88.9
CSS x TSS	–	77.3	88.3	95.7	100.0	100.0
	Adj. + noun	74.3	55.7	63.8	100.0	79.3
	Noun + noun	82.6	81.1	93.5	100.0	100.0
	Adj. + noun & noun + noun	78.8	84.5	**97.4**	100.0	88.9
CSS x TPS x TSS	–	81.7	**89.3**	94.4	100.0	100.0
	Adj. + noun	75.2	55.8	63.8	100.0	79.3
	Noun + noun	**85.4**	81.8	92.3	**100.0**	**100.0**
	Adj. + noun & noun + noun	82.4	85.2	96.1	100.0	88.9

[1] Four corpus statistic measures for calculating CSS were tested: term frequency (TF), inverse document frequency (IDF), TF-IDF, and Okapi BM25. The performance results achieved using TF were reported, because TF outperformed the others. The other measures use or partially use IDF, which frequently gave high scores to extremely-rare concept names that should not be selected as identifiers. CSS = corpus-statistic score; TPS = term-position score; TSS = term-sequence score.
[2] "–" indicates that no part-of-speech pattern was used. Adj. = adjective.
[3] ET = bridge element; DY = deficiency; DC = deficiency cause; MA = maintenance action; MM = maintenance material. The bold font indicates the highest accuracy for each concept name type.

First, the ranking function with the corpus-statistic, term-position, and term-sequence scores was effective. It achieved the highest accuracies of 85.4%, 89.3%, 100.0% and 100.0% for bridge element, deficiency, maintenance action, and maintenance material names, respectively. But, for deficiency cause names, the function with the corpus-statistic and term-sequence scores achieved the highest accuracy of 97.4%, which is 1.3% higher than that achieved using the function with all the three. This is likely because the right-hand side terms are not always the meaning-bearing terms in some deficiency cause names. For example, in the following names, the right-hand side terms are less meaningful than the those on the left-hand side: "debris buildup", "sand buildup", "poor

weld quality". The function without the term-position score, thus, achieved a higher accuracy for deficiency cause names. Second, using POS patterns for selecting identifier concept names was effective. For example, using POS patterns achieved the highest accuracies of 85.4%, 97.4%, 100.0%, and 100.0% for the bridge element, deficiency cause, maintenance action, and maintenance material names, respectively. But, for deficiency names, without using POS patterns achieved the highest accuracy of 89.3%, compared to 85.2% achieved using the "adjective + noun" and "noun + noun" patterns. This could be attributed to that some deficiency names are not noun phrases (e.g., "pulled out" and "laterally misaligned"), and restricting identifiers to noun phrases led to the decrease in the accuracy.

7.3.1.2 Performance of the Proposed Numerical Data Fusion Algorithm

Figure 7.4 shows examples of the simulation results for fusing the deficiency length measures of a patching on a girder: {12, 24, 24, 24, 24, 36, 48}, where the unit is inch. The patterns of the simulation results for fusing the numerical data in the dataset (Table 6.1) follow the same patterns shown in Figure 7.4. Overall, the results show that the proposed fusion algorithm was stable.

Two important observations were drawn from the results. First, the fusion algorithm was stable up to an uncertainty level of 2.0. As the uncertainty level increased from 2.0, the information entropy showed an increasing trend, see Figure 7.4 (a-2). The increase in the information entropy indicates that the algorithm became unstable and started to fuse the same set of deficiency measures into different intervals in a single simulation run [see the distributions of the intervals in Figure 7.4 (a-1)]. The uncertainties in numerical data negatively affect the quantification and fusion of the degree values. Due to the uncertainties, these values changed in each fusion iteration of a simulation run, which made the fusion results of the same set of data vary. Second, the algorithm was stable in the presence of the randomness of the standard deviation of the Gauss membership

function. As shown in Figure 7.4 (b-2), increasing the randomness of the standard deviation (i.e., increasing the times of randomly drawing it) did not cause change in the information entropy. This indicates that the fusion algorithm was stable and able to fuse the same set of numerical deficiency measures into the same interval [see the distributions of the interval in Figure 7.4 (b-1)]. The standard deviation was bounded by a normal distribution in the cloud model (Li et al. 2009). Despite being random, the standard deviation was always within the bound, which made it not affect the stability of the fusion algorithm.

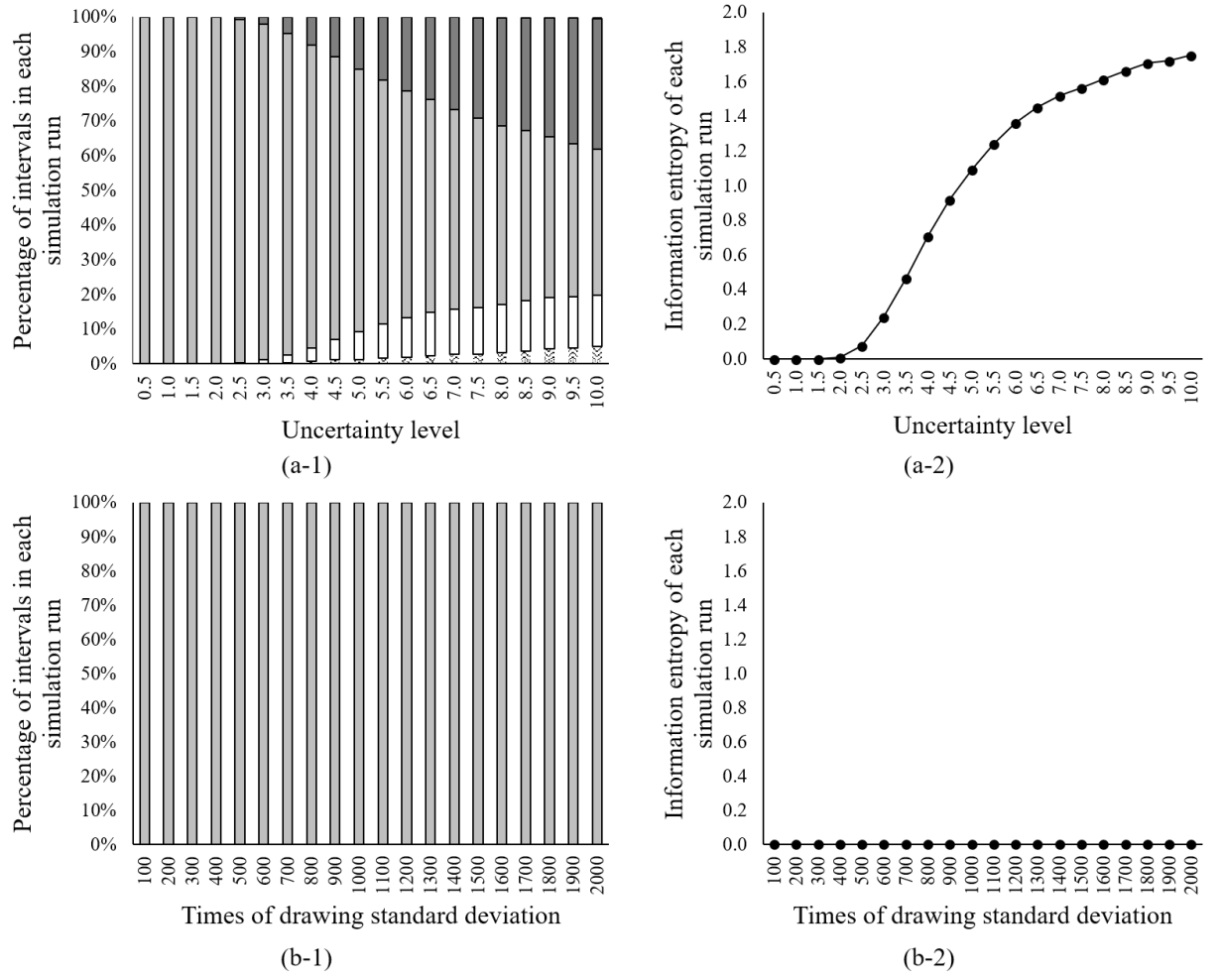


Figure 7.4. Examples of Monte Carlo simulation results for fusing multiple deficiency measures.

7.3.2 Performance Results of Method Validation

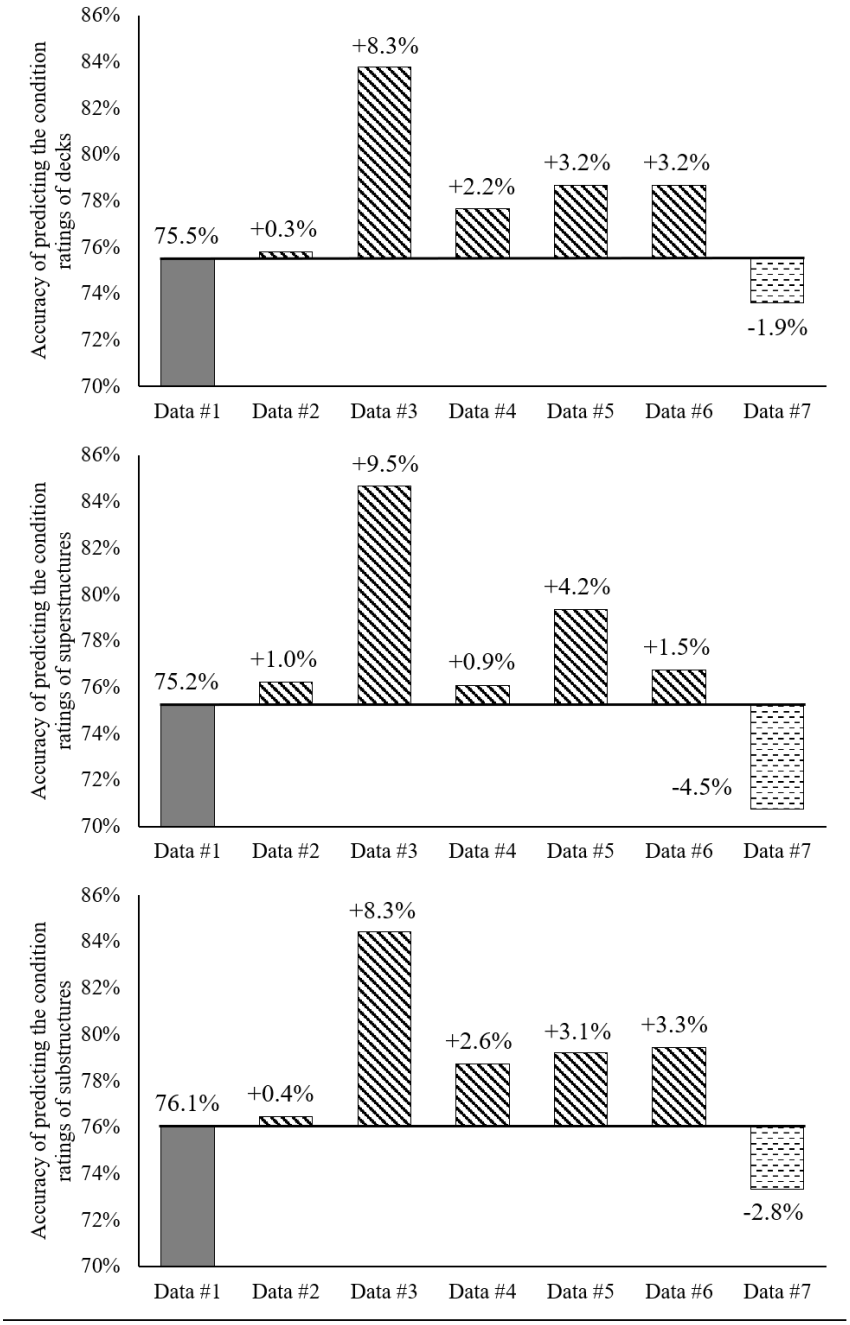
The performance results for predicting the future condition ratings of decks, superstructures, and substructures are presented in Figure 7.5. Overall, the results show that the proposed hybrid data fusion method was effective in fusing the data extracted from bridge inspection reports for supporting enhanced data-driven bridge deterioration prediction. Three important observations were also drawn from the results.

First, learning from textual bridge inspection reports, in addition to NBI data, was able to improve the prediction performance. Learning from both NBI data and the report data fused by the proposed method, compared to learning from NBI data alone, improved the prediction accuracies for decks, superstructure, and substructures by 8.3%, 9.5%, and 8.3%, respectively. NBI data, which mainly describe condition ratings and as-built characteristics of bridges, are certainly important. But, they do not include descriptions about the element-level deterioration conditions of bridges, such as those in bridge inspection reports. Such descriptions are much more detailed and dynamic in capturing the deterioration conditions of bridges in each inspection year and are, therefore, more informative in capturing the patterns of how the condition ratings evolve over time. Hence, they helped improve the performance of predicting the future ratings.

Second, data fusion is very important for learning from the data extracted from bridge inspection reports to improve the performance of bridge deterioration prediction. Learning from the unfused report data was only able to marginally improve the prediction accuracies by 0.3%, 1.0%, and 0.4%, respectively. But, learning from the report data fused by the proposed method improved these accuracies largely by 8.7% on average. The multiple – even ambiguous and conflicting – concept names and numerical data in inspection reports negatively affected the generalizability of

the machine learning models, which limited the performance of learning from bridge inspection reports in improving the prediction performance.

Third, the proposed entropy-based data fusion algorithm was effective in fusing the numerical deficiency measures for supporting the prediction. Learning from the deficiency measures fused by taking the maximum (i.e., using the measure corresponding to the worse deterioration case), the mean/total, and the variation of the measures were only able to improve the prediction accuracies by 2.7% on average. Learning from the deficiency measures fused by the combinations of the means and the variations, even, decreased the accuracies by around 3.0%, due to the doubled size of the feature dimensionality caused by the combinations. The proposed algorithm improved the accuracies by 8.7% on average, which is quite higher than the improvement rates achieved using the other methods. This is largely attributed to the fact that the proposed algorithm uses data discretization to define the interval-based representations for representing the fused measures, and utilizes information entropy to fuse deficiency measures into a single representative interval. The algorithm, thus, takes balancing the overfitting and underfitting of the machine learning prediction model and the complementarity of the measures into account, resulting in the improved prediction performance.



Note: The types of the data (data #1 to #7) were defined in Table 3. The accuracies for data #5 to #7 are the averages across different measures or combinations (see Table 3).

Figure 7.5. Performance results for predicting the future condition ratings of decks, superstructures, and substructures.

CHAPTER 8 – DATA-DRIVEN BRIDGE DETERIORATION PREDICTION

This chapter presents the proposed bridge deterioration prediction method for learning from the integrated bridge data from multiple sources (integrated as per Research Task #6) to predict the deterioration. The method development and evaluation (Research Task #7) are presented in this chapter.

8.1 Comparison to the State of the Art

At the backbone of this framework is the proposed deep learning-based bridge deterioration prediction method, which aims to learn from the integrated bridge data to predict the condition ratings of bridges and the quantities of bridge element-level deficiencies. The proposed method uses (1) manifold learning to embed the integrated data, which are of high dimensionality and sparsity, into a low-dimensional dense space; (2) recurrent neural networks to learn from the embedded data from past years to predict the conditions of bridges and their elements in the next year; and (3) cost-sensitive learning to address the class imbalance in the data to better predict the conditions. The proposed bridge deterioration prediction method is novel in two primary ways. First, it learns from both structured and unstructured bridge data from multiple sources – especially previously-untapped textual inspection reports which include a large amount of detailed data/information about bridge conditions and maintenance actions – to allow for the prediction of the condition ratings of bridges with improved performance and the prediction of the quantities of bridge element-level deficiencies. It, thus, goes beyond the current state of the art in data-driven bridge deterioration prediction, where existing methods (e.g., Morcous 2006; Chang et al. 2017; Goyal et al. 2017; Lu et al. 2019) mostly focus on learning from abstract bridge inventory data from a single source – such as the NBI data which mainly use condition ratings to describe bridge conditions – to predict, at a limited performance level, the future ratings. Second, it incorporates

manifold learning and cost-sensitive learning techniques to address the challenges of learning from highly dimensional and imbalanced data for improved performance of bridge deterioration prediction. Most of the existing methods (e.g., Huang 2010; Creary and Fang 2015; Contreras-Nieto et al. 2016; Lim and Chi 2019) leave such data challenges understudied or even untouched, which negatively affects the prediction performance and limits the ability to effectively use data to predict the deterioration.

8.2 Data-Driven Bridge Deterioration Prediction Method Development

8.2.1 Proposed Bridge Deterioration Prediction Method

To address the aforementioned knowledge gaps, a new deep learning-based bridge deterioration prediction method is proposed. It learns from the integrated bridge data, which are originally in heterogeneous formats and from multiple sources, to predict the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures) and to predict the quantities of specific bridge element-level deficiencies. The proposed method includes three primary components, as per Figure 8.1: manifold learning, recurrent neural network modeling, and cost-sensitive learning.

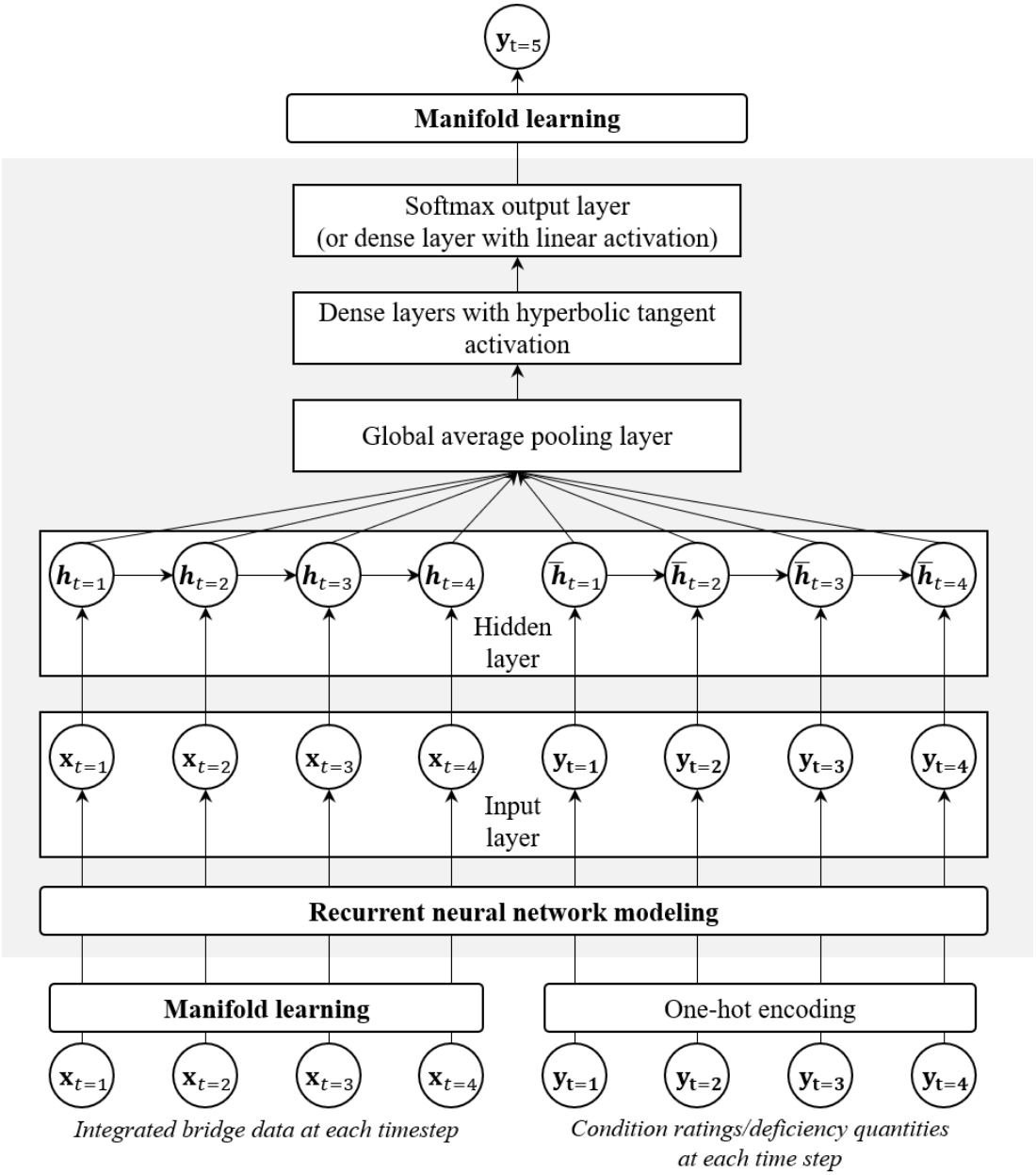


Figure 8.1. Proposed deep learning-based bridge deterioration prediction method.

8.2.1.1 Manifold Learning

Manifold learning was used to embed the integrated bridge data, which are high-dimensional and sparse, into a low-dimensional dense space for better supporting the deep learning. The isometric feature mapping (Isomap) algorithm (Tenenbaum et al. 2000) was used for the manifold learning. It was selected because of four main reasons. First, the Isomap algorithm is a nonlinear

dimensionality reduction technique that preserves the geometrical structure of data (which are often nonlinear) at all scales by mapping nearby data instances in a high-dimensional space to nearby instances in a low-dimensional space (Silva et al. 2003; Liu et al. 2008). This property allows the algorithm to represent and embed data into a low-dimensional space, in a way that the global geometrical structure of the embedded data is more faithful to that of the original data. Second, it generates a low-dimensional representation (i.e., embeddings) of the original data that is globally-optimal in a computationally-efficient way (Silva et al. 2003). Third, in most cases, it is guaranteed to converge asymptotically to the true underlying structure of the original data (Silva et al. 2003). Fourth, it is an unsupervised learning algorithm that does not require the labeling of data, which makes it more applicable to real-world applications, such as bridge deterioration prediction. Because the Isomap algorithm requires assessing the distances between data instances (which, in this research, include both numerical and categorical features), a revised Euclidean distance is proposed to allow for the distance assessment of the data instances with the mixed types of features.

Using the Isomap algorithm, embedding the integrated bridge data into a low-dimensional dense space includes three main steps. First, a neighborhood graph was constructed as the basis for approximating the geometrical structure of the original integrated data. The neighborhood graph is a graphical representation of the data, where each node of the graph is a data instance, an edge connects two data instances if one is among the k-nearest neighbors of the other, and the weight of an edge is the distance between the data instances that the edge connects. In the proposed method, $k = 10$ was adopted based on the study by Samko et al. (2006), which shows that $k \in [3, 10]$ performs well on a number of datasets. The revised Euclidean distance is proposed, as per Eq. (8.1), to allow for calculating the weights between the data instances, which have both

numerical and categorical features. In Eq. (8.1), **x** and **y** are the original data instances in the high-dimensional space, x_i and y_i are the features of their corresponding data instances, and *n* is the number of features.

Second, the shortest path-based distance matrix was constructed based on the neighborhood graph to characterize the structure of the data. The data instances that are far away from each other in the high-dimensional nonlinear space may appear close if their distances are measured using the Euclidean distance. The shortest path distances between the data instances provide a plausible approximation to their geodesic distances, which can capture the underlying distances/structures between the data instances in such a space (Fan et al. 2012). For a pair of data instances on the graph, all the paths that connect them are enumerated, and the path with the smallest distance (i.e., the total of the weights of the edges along the path) is identified as the shortest path between them. The distances of all shortest paths are used to construct the shortest path-based distance matrix $\boldsymbol{D}$.

Third, the *d*-dimensional dense embeddings of the original integrated data were formed based on the dimensions that are the most decisive of the structure of the data. Such dimensions are determined by the eigenvalues. Thus, the top *d* eigenvalues and their corresponding eigenvectors of the matrix $\boldsymbol{\tau}_G$ are used to form the *d*-dimensional dense embeddings, where $\boldsymbol{\tau}_G$ is a matrix constructed using the squared matrix of $\boldsymbol{D}$ and the centering matrix. In the proposed method, the value of *d* was set to 1200, based on the "elbow point" of the curve that shows the embedding stress against the dimensionality of the embeddings (i.e., *d*), as per Figure 8.2. The embedding stress measures the quality of the embedding (Tenenbaum et al. 2000) and was calculated using the L-2 norm between the shortest path distance-based matrix $\boldsymbol{\tau}_G$ and the Euclidean distance-based matrix $\boldsymbol{\tau}_E$.

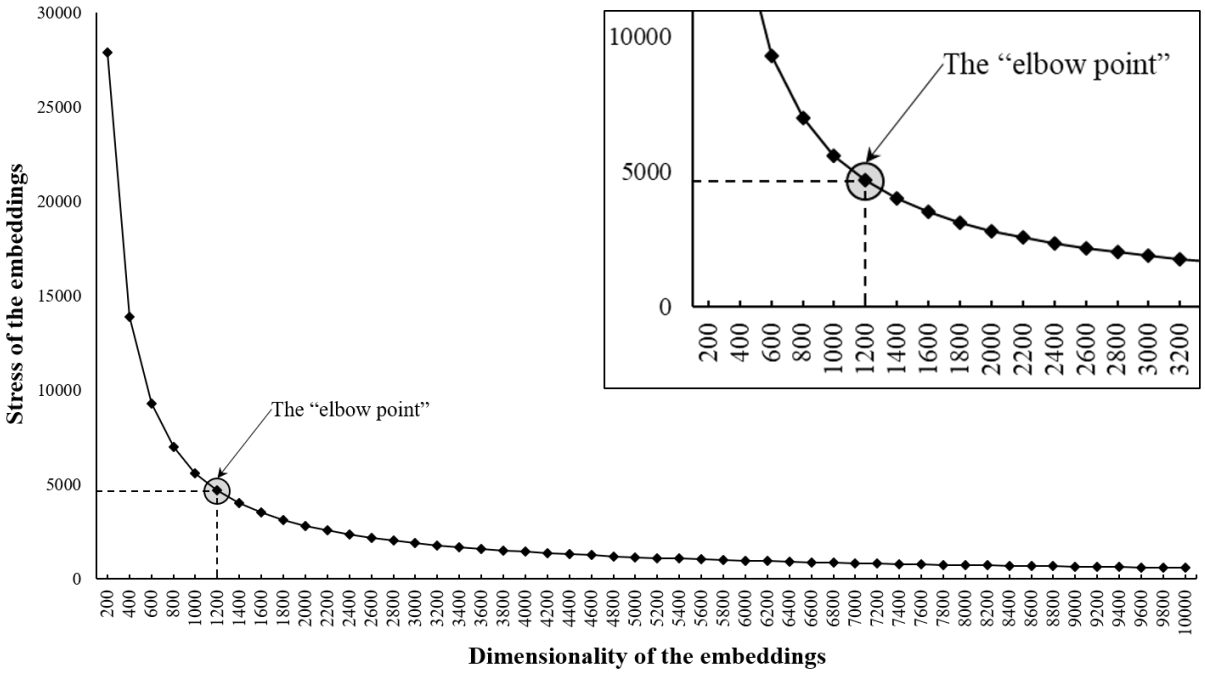


Figure 8.2. The stress of the embeddings against the dimensionality of the embeddings.

$$Distance(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_2)^2 + (x_1 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{8.1}$$

$Subject\ to$:

$$\begin{cases} x_i - y_i = x_i - y_i, & if\, x_i \text{ and } y_i \text{ are numerical features} \\ x_i - y_i = 1, & if\, x_i \text{ and } y_i \text{ are categorical features and } x_i = y_i \quad i = 1, 2, \cdots, n \\ x_i - y_i = 0, & if\, x_i \text{ and } y_i \text{ are categorical features and } x_i \neq y_i \end{cases}$$

8.2.1.2 Recurrent Neural Network Modeling

A new RNN architecture is proposed and was used to learn from the embedded bridge data from past years to predict the conditions of bridges and their elements in the next year. The RNN architecture includes an input layer, a recurrent layer, a pooling layer, a set of nonlinear dense layers, and an output layer, as per Figure 8.1. The RNN architecture was modeled in this way for three main reasons. First, it uses hidden states to capture and remember the temporal dynamics that connect data derived from a physical system over time (Che et al. 2017), which allows for capturing the patterns of the sequential changes of the bridge conditions over the past years and

leveraging the patterns to better predict the conditions of bridges in the next year. Second, it is composed of high-dimensional hidden states with nonlinear dynamics (Salehinejad et al. 2017), which allows for better capturing the dimensionality and the nonlinearity of the data. Third, compared to the RNN with long short-term units or gated recurrent units, it can achieve comparable performance and improve computational efficiency, when dealing with data that have short sequences and short-range dependencies (Tang et al. 2019), such as the bridge data.

The input layer takes two types of inputs: the embedded bridge data and the target classes from past years. The embedded data are represented as a sequence of vectors $\{\mathbf{x}_{t=1}, \mathbf{x}_{t=2}, \cdots \mathbf{x}_{t=n}\}$, where $\mathbf{x}_{t=n}$ is an instance of the embedded data of a bridge at inspection year $t = n$, $\mathbf{x}_{t=n} = \left(x_{t=1}^{1}, x_{t=1}^{2}, \cdots x_{t=1}^{d}\right)$, $x_{t=1}^{d}$ is the d^{th} feature of the instance, and d is the size of the embeddings. The target classes are represented as a sequence of vectors $\{\mathbf{y}_{t=1}, \mathbf{y}_{t=2}, \cdots \mathbf{y}_{t=n}\}$, where $\mathbf{y}_{t=n}$ is a one-hot encoded vector of the target class of the bridge at inspection year $t = n$ and a target class could be the condition rating of a primary bridge component or the (discretized) quantity of a specific bridge element-level deficiency. The target class information from past years $\mathbf{y}_{t=n}$ is included in the embedded data $\mathbf{x}_{t=n}$, but it was separately modeled as input because of two reasons. First, the target classes from previous years are directly related to the target class to be predicted in the next year (i.e., $\mathbf{y}_{t=n+1}$) and they are, thus, very informative and important to the prediction. Second, although the integrated data capture the information about the target classes, the manifold learning might have embedded the data into a low-dimensional space that makes the importance of these target classes less apparent.

The recurrent layer contains two sets of hidden states, each of which corresponds to a type of input. A hidden state takes the values mapped from its corresponding input vector/node in the input layer and the values mapped from its previous state (in the same set). The mapping is conducted using

Eq. (8.2) (Graves et al. 2013), where $\mathbf{h}_{t=i}$ is a hidden vector containing the mapped values for a hidden state at timestep $t = i$, $tanh$ is the hyperbolic tangent activation function, $\mathbf{W}_{IH}$ is a weight matrix for the input-to-state mapping, $\mathbf{W}_{HH}$ is a weight matrix for the state-to-state mapping, $\mathbf{b}_h$ is a bias vector of hidden states, and $\mathbf{z}_{t=i}$ could be $\mathbf{x}_{t=i}$ or $\mathbf{y}_{t=i}$. The hyperbolic tangent activation function was selected because it is commonly used in RNN and its second derivative can sustain for a long range before getting to zero to avoid potential vanishing gradients.

$$\boldsymbol{h}_{t=i} = tanh(\boldsymbol{W}_{IH}\boldsymbol{z}_{t=i} + \boldsymbol{W}_{HH}\boldsymbol{z}_{t=i} + \boldsymbol{b}_h), \qquad i = 1,2,\cdots n \tag{8.2}$$

The pooling layer is a global average pooling layer that takes all the hidden vectors as input, and outputs a single vector with the same size of the hidden vectors. It aims to reduce the dimensionality of the intermediate representations of the data (i.e., the number of the hidden vectors) and reduce the sensitivity of the output to noise. To represent the hidden vectors using a single vector, the pooling layer uses the average of all the values that are at the same position of all the hidden vectors to represent these values in the single vector. For example, the first value in the single vector is the average of all the first values of the hidden vectors. A set of nonlinear dense layers with the hyperbolic tangent activation function was added upon the pooling layer to capture the patterns of the single hidden vector. The dense layer was connected to the output layer for making a final prediction. A softmax output layer was used to predict the (categorical) condition rating of a primary component of a bridge, and a dense output layer with the linear activation function was used to predict the (numerical) quantity of a bridge element-level deficiency.

8.2.1.3 Cost-Sensitive Learning

Cost-sensitive learning was used to address the imbalance in the bridge data to better predict the condition ratings of the primary bridge components. In the proposed method, the binary focal loss

function (Lin et al. 2017) was adopted, and was extended to a multi-class loss function for the cost-sensitive learning. This function was chosen for extension for two main reasons. First, unlike sampling-based approaches for dealing with data imbalance, the use of this function does not increase or decrease the number of data instances, which helps avoid overfitting and the loss of important instances. Second, it uses a modulating factor that directly considers the costs of misclassifications to better deal with the skewed distributions of the data classes. The modulating factor is, thus, more effective in dealing with the imbalance, compared to the weighting factor (i.e., inverse class frequency) used in existing cost-sensitive learning methods.

Because the binary focal loss cannot deal with multi-class classification problems, it was extended to a multi-class focal loss based on the multi-class cross-entropy loss. The multi-class cross-entropy loss is calculated using Eq. (8.3) (De Boer et al. 2005), where C is the number of classes (e.g., the number of different condition ratings of decks), $I_{i,c}$ is a binary indicator, $I_{i,c} = 1$ if the c^{th} class if the correct class/label for the i^{th} data instance and $I_{i,c} = 0$ otherwise, and $p_{i,c}$ is the probability of the i^{th} data instance being classified into the c^{th} class. The cross-entropy loss treats the cost of misclassifying the instances in the minority classes and the cost of misclassifying the instances in the majority classes equally, without addressing the imbalance. The modulating factor defined in the binary focal loss was added into the cross-entropy loss, resulting in a multi-class focal loss that can deal with both multi-class classification problems and imbalance in data, as per Eq. (8.4), where $\left(1 - p_{i,c}\right)^{\gamma}$ is a modulating factor for adjusting the costs of misclassifications, $\gamma = 2$ [based on Lin et al. (2017)], and the other notations follow those defined in Eq. (8.3).

The proposed multi-class focal loss function serves as the objective function for training the RNN (i.e., the training process aims to minimize the multi-class focal loss). It exhibits the following

properties, which help adjust the two types of misclassification costs for dealing with the imbalance. First, it decreases the contributions of the data instances in the majority classes to the training of the RNN. These data instances are relatively easy to be correctly classified due to the large percentage of them in the dataset. In this case, the probabilities of such instances being classified into their correct classes approach to 1 quickly. This decreases the values of the modulating factor for these instances and, thus, exponentially decreases their misclassification costs. As a result, the RNN focuses less on learning from the data instances in the majority classes, because the correct classifications of them do not help reduce the overall loss/cost much. Second, it increases the contributions of the data instances in the minority classes to the training of the RNN. These data instances are hard to be correctly classified due to the small percentage of them in the dataset. In this case, the probabilities of such instances being classified into their correct classes are far away from 1. This increases the values of the modulating factor for these instances, and thus, exponentially increases their misclassification costs. As a result, in order to reduce the overall loss/cost, the RNN focuses more on learning from the data instances in the minority classes and tuning its parameters to adjust the classifier for better separating such instances.

$$Cross\text{-}entropy\ loss = -\sum_{c=1}^{C} I_{i,c} \times \log\left(p_{i,c}\right) \tag{8.3}$$

$$Multi\text{-}class\ focal\ loss = -\sum_{c=1}^{C} \left(1 - p_{i,c}\right)^{\gamma} \times I_{i,c} \times \log\left(p_{i,c}\right) \tag{8.4}$$

8.2.2 Implementation of the Proposed Method

The proposed deep learning-based bridge deterioration prediction method was implemented in predicting the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures) and in predicting the quantities (i.e., the number of deficiency instances and the total

length/area of the deficiency instances) of three common types bridge element-level deficiencies – pier spall, deck delamination, and girder crack. The implementation included four primary steps: data preparation, information extraction and data integration, algorithm training, and performance evaluation. An overview of the implementation methodology is presented in Figure 8.3. A step-by-step illustration of the implementation methodology is presented in Figure 8.4.

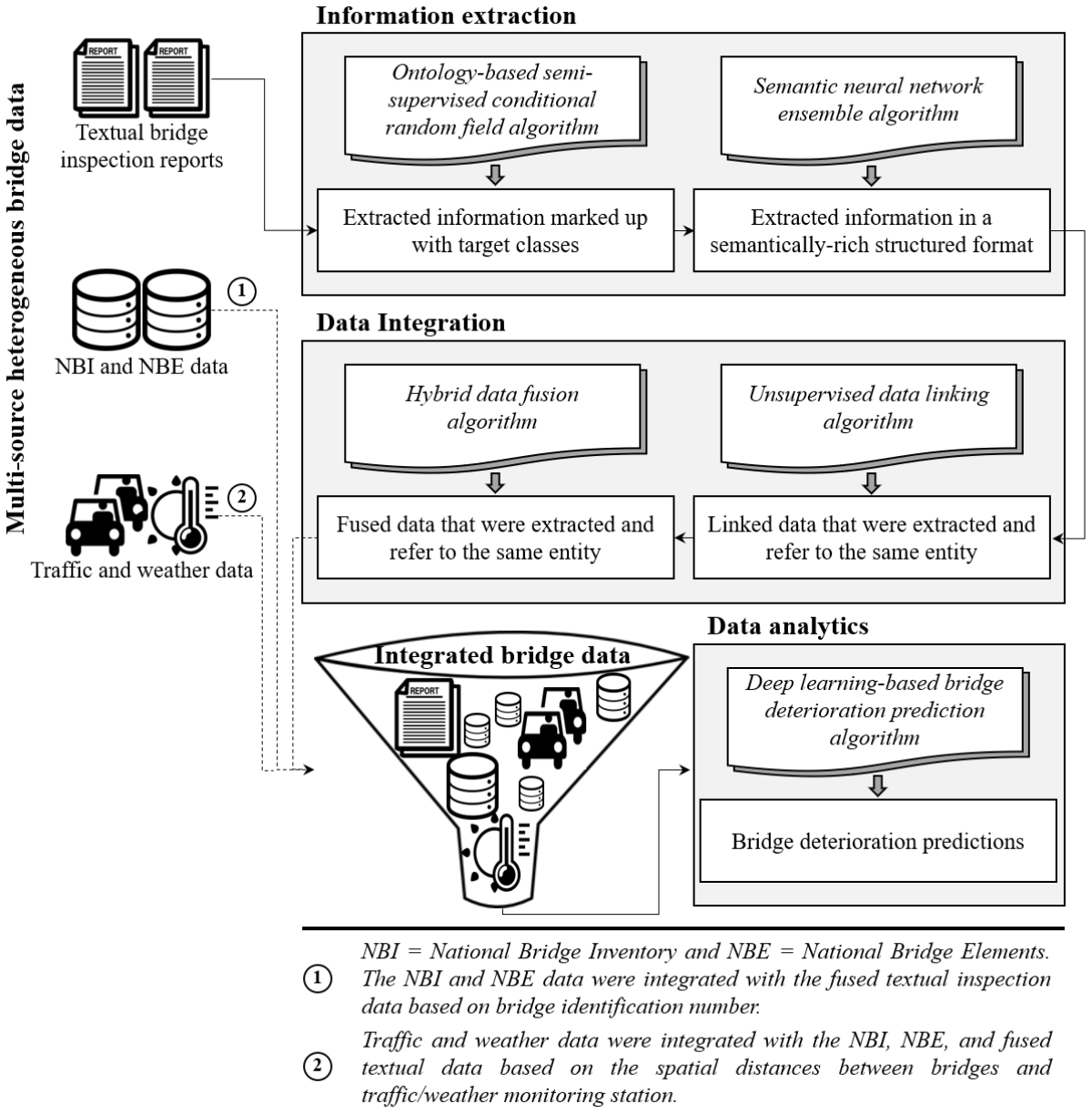


Figure 8.3. Implementation methodology for the proposed bridge deterioration prediction approach (the proposed bridge data analytics framework).

- **Original text from a bridge inspection report (partial)**

Column 2B has a patch on the NW corner near the groundline. At SW corner, behind the thrie beam, bridge rail has an 18" x 6" x 3" deep edge spall. The north concrete bridge rail at the east end has a 3 ft. x 6" x 3" deep top edge spall.

1: Information extraction

- **Extracted information represented in a semantically-rich structured format (partial)**

1. <ET = "column", DY = "patch", DC = N/A, NM = N/A, NU = "N/A">
2. <ET = "bridge rail", DY = "deep edge spall", DC = N/A, NM = "18, 6, 3", NU = "inch, inch, inch">
3. <ET = "north concrete bridge rail", DY = "deep top edge spall", DC = N/A, NM = "3, 6, 3", NU = "feet, inch, inch">

* ET = bridge element; DY = deficiency; DC = deficiency cause; NM = numerical measure; NU = numerical measure unit.

2: Data integration (linking and fusing the data extracted from the report)

- **Fused data that were extracted and refer to the same entity (partial)**

1. <ET = "column", DY = "patch", DC = N/A, NM = N/A, NU = "N/A">
2. <ET = "concrete bridge rail", DY = "deep spall", DC = N/A, NM = "[16, 34), [5, 10), [1, 5)", NU = "inch, inch, inch", number of deficiency instances = 2>

* The second and third records from the information extraction were linked and fused into a single record in a unified representation.

3: Data integration (integrating the fused data with the other types of structured bridge data)

- **Integrated heterogeneous bridge data from multiple sources (partial)**

- Year_built (NBI) = 1956
- Deck_condition_rating (NBI) = 6
- Superstructure_condition_rating (NBI) = 7
- Substructure_condition_rating (NBI) = 6
- Concrete_deck_quantity (NBE) = 2,432
- Concrete_deck_state_1_quantitiy (NBE) = 2,432
- Concrete_deck_state_1_quantitiy (NBE) = 0
- Concrete_deck_state_1_quantitiy (NBE) = 0
- Concrete_deck_state_1_quantitiy (NBE) = 0
- Column_patch (report) = Yes
- Concrete_bridge_rail_deep_spall (report) = Yes
- Concrete_bridge_rail_deep_spall_length (report) = [16, 34)
- Concrete_bridge_rail_deep_spall_width (report) = [5, 10)
- Concrete_bridge_rail_deep_spall_depth (report) = [1, 5)
- Average_daily_traffic (traffic) = 20,255
- Percentage_of _single_unit_trucks (traffic) = 3.78%
- Percentage_of _double_unit_trucks (traffic) = 1.76%
- Percentage_of _triple_unit_trucks (traffic) = 0.18%
- Annual_average_wind_speed (weather) = 5.95
- Cooling_degree_days (weather) = 352
- Monthly_mean_minimum_temperature (weather) = 40.29
- Monthly_mean_maximum_temperature (weather) = 58.39

* The NBI and NBE data were integrated with the fused textual report data based on structure identification number. They were further integrated with the traffic and weather data based on latitude and longitude (i.e., the distance between the bridge and the traffic/weather monitoring station; NBI = National Bridge Inventory; NBE = National Bridge Elements.

4: Data analytics (learning from the integrated bridge data to predict bridge deterioration)

- **Input:** the integrated bridge data from previous years (e.g., 2009, 2011, 2013, and 2015)
- **Output:** predicted conditions of the bridge in 2017
  - Predicted condition ratings of the deck (5), superstructure (7), and substructure (6).
  - Predicted quantities of the deck spall: total spall instances = 1 and total spall length = 4 inch.

Figure 8.4. An illustrative example of the implementation methodology of the proposed deep learning-based bridge deterioration prediction method.

8.2.2.1 Data Preparation

A dataset, which contains the NBI and NBE data, the traffic and weather data, and the textual bridge inspection reports of 2,646 state-owned bridges in the state of Washington, was created.

The details of the created dataset are summarized in Table 8.1. The data from 2006-2015 were used for predicting the conditions of the bridges in 2016 and 2017. The datasets were further split into training and testing sets using 3-fold cross validation, where the ratio of the number of bridges in a training set to the number of bridges in a testing set is 2:1. In addition, the data of the year for which the prediction is made (e.g., 2017) were excluded from both sets to avoid data leakage (i.e., making predictions using data that are not available in practice at the time of the prediction). For a full list of the features of the data, the readers are referred to Appendix A.

In order to evaluate the error propagation across the different data analytics steps (Section 8.3.3), an additional gold standard was developed to identify the error rates of the information extraction, data linking, and data fusion algorithms (Section 8.2.2.2) for comparative purposes. The gold standard for each data analytics step was, separately, prepared by three annotators. The three are researchers with background in civil engineering, natural language processing, and machine learning. Disagreements were resolved using discussion to reach consensus and full annotator agreement for the final gold standard.

Table 8.1. Details of the created dataset.

Data	Description	Year range [1]	Source
National Bridge Inventory (NBI) data	The NBI data are bridge-level data. They include features about the locations of bridges (e.g., highway agency district, longitude, and latitude), the geometric, structural, and construction characteristics of bridges (e.g., bridge length, deck width, design load, functional classification, year built), and the conditions of bridges (e.g., the condition ratings of the primary bridge components – decks, superstructures, and substructures).	2006-2017	Federal Highway Administration (FHWA 2019)
National Bridge Elements (NBE) data	The NBE data are element-level data. They include the total quantity of a bridge element and the quantities of the element in four condition states: "Good", "Fair", "Poor", and "Severe". The NBE data include the bridge elements that are the national bridge elements and the agency developed elements. At the highest level of abstraction, the NBE data include bridge elements, such as decks, superstructures, substructures, bearings, approach slabs, overlays, etc. Under these abstract elements, a total of 185 detailed bridge elements (e.g., concrete decks, steel orthotropic decks, concrete stringers, concrete trusses) are defined in the NBE data.	2006-2017	The Bridge Engineering Information System of the Washington Department of Transportation (WSDOT) [2]
Traffic data	The traffic data include features about the average daily traffic and the percentages of single, double, and triple unit trucks. Two types of traffic data were included in the dataset: original and interpolated. For example, for the average daily traffic of a bridge, the original counts of the average daily traffic from three traffic monitoring stations that are the closest to the bridge and the interpolated count based on the three original counts were included. The interpolation was conducted using the barycentric interpolation method. For the years when no traffic data are available, the averages of the available traffic data were used as substitutes.	2005-2017	The Transportation Data, GIS & Modeling Office of the WSDOT (WSDOT 2019)
Weather data	The weather data include a total of 49 features, such as cooling degree days computed with bases of 45, 50, 55, 57, 60, 65, 70, and 72 ^{o}F, diurnal temperature range, heating degree days, precipitation totals, snowfall totals, and average, maximum temperature, minimum temperate, etc. Similar to the traffic data, both original and interpolated weather data were included.	2006-2017	National Oceanic and Atmospheric Administration (NOAA 2019)
Textual bridge inspection reports	The textual bridge inspection reports include technically-detailed data/information about bridge conditions (e.g., the types of deficiencies and the quantities of deficiencies) and maintenance actions (e.g., the types of maintenance actions and the types of maintenance material).	2006-2017	The Bridge Engineering Information System of the WSDOT

[1] Bridges are inspected at a 2-year interval. For a bridge, its data could be in the year range from 2006 to 2016 or from 2007 to 2017.
[2] The NBE data are from the textual bridge inspection reports, collected from the WSDOT.

8.2.2.2 Information Extraction and Data Integration

Information extraction was conducted to extract information from the unstructured textual bridge inspection reports and represent the extracted information in a semantically-rich structured way. The following types of information that describe bridge conditions and maintenance actions were extracted from the reports: “bridge element”, “deficiency”, “deficiency cause”, “maintenance action”, “maintenance material”, “numerical measure”, “numerical measure unit”, “categorical quantity measure”, “categorical severity measure”, and “date”. The ontology-based semi-supervised conditional random field algorithm (developed as per Research Task #3, in Chapter 4) was used for extracting the information from the reports. The semantic neural network ensemble algorithm (developed as per Research Task #4, in Chapter 5) was used extracting the dependency relations from the text for representing the extracted information in a structured way. For example, as per Figure 8.4, the information was extracted from the following sentence and was then represented in a structured format <bridge element = “bridge rail”, deficiency = “deep edge spall”, deficiency cause = N/A, numerical measure = “18, 6, 3”, numerical measure unit = “inch, inch, inch”>: “At SW corner, behind the thrie beam, bridge rail has an 18” x 6” x 3” deep edge spall.” (WSDOT 2015).

Data integration was conducted to link and fuse the records that were extracted from the reports and refer to the same entity (e.g., the same type of deficiency on the same type of bridge element). The spectral clustering-based data linking algorithm (developed as per Research Task #5, in Chapter 6) was used for linking the records, and the hybrid data fusion algorithm (developed as per Research Task #6, in Chapter 7) was used for fusing the linked records into a unified representation. For example, as per Figure 8.4, the following two records are referring to the same entity (i.e., the spall on the bridge rail) and were thus linked: <bridge element = “bridge rail”,

deficiency = “deep edge spall”, deficiency cause = N/A, numerical measure = “18, 6, 3”, numerical measure unit = “inch, inch, inch”> and <bridge element = “north concrete bridge rail”, deficiency = “deep top edge spall”, deficiency cause = N/A, numerical measure = “3, 6, 3”, numerical measure unit = “feet, inch, inch”>. The linked records were then fused into a single record in a unified representation: <bridge element = “concrete bridge rail”, deficiency = “deep spall”, deficiency cause = N/A, numerical measure = “[16, 34), [5, 10), [1, 5)”, numerical measure unit = “inch, inch, inch”, number of deficiency instances = 2>, where the multiple concept names (e.g., “bridge rail” and “north concrete bridge rail”) were fused into a canonical name (i.e., “concrete bridge rail”) and the multiple numerical deficiency measure data (e.g., 18 inch vs. 3 ft) were fused into a single representative interval-based representation (i.e., the representative interval for the multiple deficiency measures = [16, 34) and the number of the deficiency instances = 2).

Data integration was also conducted to integrate the fused report data with the other types of structured bridge data. Prior to the integration, the missing values of the numerical features of the structured data – the NBI and NBE as well as the traffic and weather data – were imputed using the mean of the corresponding feature values that are available. The missing values of the categorical features were represented using a dummy value “N/A”. The integration included the following two steps. First, the fused report data were integrated with the NBI data and NBE data based on the structure identification numbers of the bridges. Second, the data integrated in the first step were further integrated with the traffic data and weather data based on the spatial distances between the bridges and the traffic/weather monitoring stations. For example, for a bridge, the top-three closet traffic monitoring stations to the bridge were selected, and the original traffic data from these stations and the data interpolated based on the original data (using the barycentric interpolation method) were integrated with the NBI, NBE, and fused data of the bridge.

8.2.2.3 Algorithm Training

The proposed prediction algorithm was trained using the integrated bridge data in the training set. The training included four main steps. First, the integrated bridge data were embedded into a low-dimensional dense space using the Isomap algorithm. The size of the embeddings was set to 1200 based on Figure 8.2. The algorithm was implemented using the manifold learning module in the scikit-learn package (Pedregosa et al. 2011). Second, the proposed RNN architecture (as per Figure 8.1) was modeled, and the multi-class focal loss function [as per Eq. (8.4)] for predicting the condition ratings and the mean square error loss function for predicting the quantities of deficiencies were defined. The model and loss functions were implemented in a Python program developed using Keras (Chollet 2015), which is a Python deep learning library. Third, the hyperparameters of the algorithm were set based on hyperparameter tuning, as follows: size of hidden states = 128, batch size = 32, number of dense layers = 6, maximum number of epochs = 1000, quantity monitored for early stopping = training loss, minimum change in the monitored quantity = 0, and patience = 25. Fourth, the embedded data were fed into the RNN architecture to start the training process, and the training proceeded until the early stopping/convergence criterion was met (i.e., the change of the training loss remained as 0 for 25 epochs).

8.2.2.4 Performance Evaluation

The performance of the proposed method in predicting the condition ratings and in predicting the quantities of bridge element-level deficiencies was, separately, evaluated. In the first evaluation case, the performance was measured using two metrics: macro-precision and macro-recall. Macro-precision and macro-recall measure the overall performance using the mean of the precision and recall for each category, respectively. The two metrics were calculated using Eqs. (8.5) and (8.6) (Madjarov et al. 2012), respectively, where tp_i = number of condition ratings predicted correctly

as positive of a condition rating category i; fp_i = number of condition ratings predicted incorrectly as positive of category i; fn_i = number of condition ratings predicted incorrectly as negative of category i; C = total number of condition ratings categories that a primary bridge component could have; $(tp_i + fp_i)$ = total number of predicted condition ratings for category i; and $(tp_i + fn_i)$ = total number of true condition ratings for category i.

$$Macro\text{-}precision = \frac{1}{C}\sum_{i=1}^{c}\frac{tp_i}{tp_i + fp_i} \tag{8.5}$$

$$Macro\text{-}precision = \frac{1}{C}\sum_{i=1}^{c}\frac{tp_i}{tp_i + fn_i} \tag{8.6}$$

In the second evaluation case, the performance was measured using three metrics: root mean square error (RMSE), coefficient of variation (CV), and coefficient of determination (R^2). The three metrics were calculated using Eqs. (8.7) to (8.9) (Chai and Draxler 2014; Abdi 2010; Nagelkerke 1991), respectively, where $y_{predict,j}$ is the predicted quantity of a deficiency for the j^{th} bridge, which could be the number of deficiency instances of the same type on a bridge element of the same type or the total length/area of the deficiency instances; $y_{data,j}$ is the actual quantity of the deficiency; $\bar{y}_{data}$ is the average of the actual quantities of the deficiencies of the same type across all the bridges in the dataset, and N is the number of bridges in the dataset. RMSE measures, on average, how concentrated the predicted data are around the line that best fits the actual data. CV measures the extent to which the overall prediction error varies with respect to the mean of the actual data. R^2 measures the percentage of the variance of the actual data explained by the prediction model.

$$Root\ mean\ square\ error = \sqrt{\frac{\sum_{j=1}^{N}\left(y_{predict,j} - y_{data,j}\right)^2}{N}} \tag{8.7}$$

$$Coefficiency\ of\ variation\ (\%) = \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_{predict,j} - y_{data,j}\right)^2}}{\bar{y}_{data}} \times 100 \quad (8.8)$$

$$Coefficient\ of\ determination = 1 - \frac{\sum_{j=1}^{N}\left(y_{predict,j} - y_{data,j}\right)^2}{\sum_{j=1}^{N}\left(y_{data,j} - \bar{y}_{data}\right)^2} \quad (8.9)$$

8.3 Data-Driven Bridge Deterioration Prediction Method Evaluation

8.3.1 Performance of Predicting Bridge Condition Ratings

The performance of the proposed method was compared to those of two baseline methods to evaluate the effectiveness of the proposed method for addressing the imbalance in the data and the effectiveness of learning from integrated bridge data from multiple sources. The first baseline learned from the integrated multi-source bridge data, using the proposed RNN architecture (as per Figure 8.1), but with the cross-entropy loss function. Unlike the proposed multi-class focal loss function, the cross-entropy loss function treats the cost of misclassifications in the minority classes and the cost of misclassifications in the majority classes equally and, hence, does not address the imbalance in the data. The second baseline is same as the first (i.e., used the RNN architecture with the cross-entropy loss function), but only learned from the NBI data. The second baseline method is similar to existing data-driven bridge deterioration prediction methods, which mostly focus on learning from single-source bridge inventory data (e.g., NBI data or similar inventory data collected by different countries), without addressing data imbalance. The performance results of these methods for predicting the condition ratings for decks, superstructures, and substructures are summarized in Tables 8.2 to 8.4, respectively. The averages of the performance results of the three methods, across the three component types, are shown in Table 8.5. The breakdown of the performance results of the proposed method for different bridge types are shown in Table 8.6. Three main conclusions were drawn from these results.

First, the proposed method was effective in dealing with the imbalance in the data. For example, the proposed method achieved a macro-precision and macro-recall of 89.1% and 83.7%, 91.3% and 85.9%, and 89.3% and 87.8%, when predicting the condition ratings for the decks, superstructures, and substructures, respectively. Without addressing the imbalance, the first baseline method only achieved a macro-precision and macro-recall of 81.9% and 75.6%, 82.2% and 67.8%, and 83.2% and 72.4%, respectively. The performance improvements were achieved mainly because the method used a proposed multi-class focal loss function. The loss function increased the cost for the misclassifications of the data instances in the minority classes and decreased the cost for the misclassifications of the data instances in the majority classes. This made the proposed method focus more on learning from the instances in the minority classes and, thus, largely improved the performance for such classes, which led to the improvements in the overall performance. For example, compared to the first baseline method, the proposed method barely improved the performance for the majority classes when predicting the condition ratings of the decks, as per Table 8.2. For the majority class of condition rating category "7", it did not improve the precision (95.1% vs. 95.4%) and the recall (94.7% vs. 95.1%). But the proposed method largely improved the performance for the minority classes. For the minority class of condition rating category "4", it improved the precision by 17.0% (86.4% vs. 69.4%) and the recall by 36.2% (78.7% vs. 42.5%).

Second, learning from the integrated bridge data outperformed only learning from the NBI data in predicting the condition ratings of the primary bridge components. For example, learning from the NBI data alone, the second baseline method only achieved a macro-precision and macro-recall of 75.6% and 65.1%, 75.8% and 62.6%, and 73.3% and 62.5%, when predicting the condition ratings for the decks, superstructures, and substructures, respectively. On average, as per Table 8.5, the

precisions are 15.0% lower and recall are 23.3% lower, compared to those achieved by the proposed method, respectively. This is mainly attributed to the fact that abstract bridge inventory data (e.g., the NBI data) mainly include features about the as-built geometric, structural, and construction characteristics of the bridges and mainly describe the conditions of the bridges by condition ratings, only. Such abstract data, although are very useful and important, are not enough in capturing the patterns of the sequential changes of the condition ratings (e.g., the condition rating of a bridge in the current year is the same as that in the past year, but is worsened in the next year due to exacerbated deficiencies), because they lack technically-detailed data about bridge conditions (e.g., the types of bridge element-level deficiencies, their quantities, and causes) and maintenance actions (e.g., the types of maintenance actions and material). Depending on the deficiency conditions of a bridge and the kind of maintenance it received, the condition ratings of bridges that have same/similar as-built characteristics could be in different rating categories. In this case, only learning from NBI data was not able to sufficiently capture the patterns of the condition ratings and was, thus, limited in predicting the ratings in the next year. Conversely, learning from the integrated multi-source bridge data, especially the detailed data about the deficiency conditions and the maintenance actions for the bridge elements, allowed the prediction method to sufficiently capture the patterns for better differentiating different deterioration cases for improved prediction performance.

Third, the proposed method achieved relatively higher performances for bridge types with less variability in condition ratings, rather than bridge types with a larger size of instances. For example, although the total number of timber bridges in the dataset is much smaller than that of the prestressed concrete bridges (93 vs. 1,147, as per Table 8.6), the proposed method achieved a higher performance for the timber bridges, when predicting the condition ratings of the decks

(precision = 92.4% and recall = 89.4% vs. precision = 89.0% and recall = 83.4%), the superstructures (precision = 95.0% and recall = 90.8% vs. precision = 91.6% and recall = 87.5%), and the substructures (precision = 92.6% and recall = 93.5% vs. precision = 90.8% and recall = 86.9%). This is largely attributed to the smaller variability in the condition ratings of timber bridges – the standard deviation of the bridge numbers in different deck, superstructure, and substructure rating categories is 19.59, 19.11, and 24.40, compared to that of 244.97, 245.50, and 289.43 for the prestressed concrete bridges, respectively. The results indicate that data variability, not only data size, is essential for improved prediction performance.

Table 8.2. The performance results of the proposed method and the baseline methods for predicting the condition ratings of the decks.

CR [1]	TP + FN	Proposed method [2]				Baseline method #1 [3]				Baseline method #2 [4]			
		TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)
"N"	67	67	69	97.2	100.0	67	69	97.2	100.0	67	69	97.2	100.0
"8"	64	45	50	90.0	70.5	43	49	88.1	67.4	35	43	81.4	55.7
"7"	1596	1512	1590	95.1	94.7	1518	1592	95.4	95.1	1424	1515	94.0	89.2
"6"	747	685	781	87.7	91.7	690	784	88.0	92.4	688	903	76.2	92.1
"5"	66	49	62	79.3	74.2	42	69	61.0	63.6	35	47	74.6	53.0
"4"	56	44	51	86.4	78.7	24	35	69.4	42.5	21	32	65.7	37.6
"3"	50	38	43	88.2	75.9	34	46	73.9	68.0	14	35	40.1	28.2
Macro-precision/macro-recall				89.1	83.7			81.9	75.6			75.6	65.1

[1] CR = condition rating category; "N" = "not applicable"; "8" = "very good condition"; "7" = "good condition"; "6" = satisfactory condition"; "5" = "fair condition"; "4" = "poor condition"; "3" = "serious condition".

[2] The proposed deep learning-based bridge deterioration prediction method, which uses the proposed RNN architecture (as per Fig, 1) for learning from the integrated data, the Isomap algorithm for data embedding, and the proposed multi-class focal loss function for addressing the imbalance in the data. It learns from the integrated bridge data from multiple sources.

[3] The first baseline method, which uses the RNN architecture, the Isomap algorithm, and the multi-class cross-entropy loss function (without addressing the imbalance). It learns from the integrated bridge data from multiple sources.

[4] The second baseline method, which uses the RNN architecture and the multi-class cross-entropy loss function (without addressing the imbalance). It only learns from the National Bridge Inventory (NBI) data.

[*] The other notations in the table: TP = true positives; FN = false negatives; FP = false positives; P = precision; R = recall; "–" = not applicable. The numbers of TP and TP + FP for each condition rating category are the sums achieved in the 3-fold cross validation, respectively. The precision and recall for each condition rating category are the averages achieved in the 3-fold cross validation, respectively. As a result, for each category, TP/(TP + FP) might not be exactly equal to the precision, and TP/(TP + FN) might not be exactly equal to the recall.

Table 8.3. The performance results of the proposed method and the baseline methods for predicting the condition ratings of the superstructures.

CR [1]	TP + FN	Proposed method [2]				Baseline method #1 [3]				Baseline method #2 [4]			
		TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)
"N"	54	54	56	96.5	100.0	54	56	96.5	100.0	54	56	96.5	100.0
"8"	60	46	54	85.5	76.7	42	56	75.2	70.0	42	56	75.2	70.0
"7"	1536	1476	1518	97.2	96.1	1476	1515	97.4	96.1	1476	1515	97.4	96.1
"6"	825	785	864	90.9	95.2	758	873	86.8	91.9	758	879	86.2	91.9
"5"	145	112	132	85.0	77.3	97	128	76.2	66.9	97	130	74.9	66.9
"4"	22	16	19	84.1	72.6	11	18	61.3	50.0	3	10	24.4	13.7
"3"	4	3	3	100.0	83.3	0	0	–	0.0	0	0	–	0.0
Macro-precision/macro-recall				91.3	85.9			82.2	67.8			75.8	62.6

[1] CR = condition rating category; "N" = "not applicable"; "8" = "very good condition"; "7" = "good condition"; "6" = satisfactory condition"; "5" = "fair condition"; "4" = "poor condition"; "3" = "serious condition".
[2] The proposed deep learning-based bridge deterioration prediction method, which uses the proposed RNN architecture (as per Fig, 1) for learning from the integrated data, the Isomap algorithm for data embedding, and the proposed multi-class focal loss function for addressing the imbalance in the data. It learns from the integrated bridge data from multiple sources.
[3] The first baseline method, which uses the RNN architecture, the Isomap algorithm, and the multi-class cross-entropy loss function (without addressing the imbalance). It learns from the integrated bridge data from multiple sources.
[4] The second baseline method, which uses the RNN architecture and the multi-class cross-entropy loss function (without addressing the imbalance). It only learns from the National Bridge Inventory (NBI) data.
[*] The other notations in the table: TP = true positives; FN = false negatives; FP = false positives; P = precision; R = recall; "–" = not applicable. The numbers of TP and TP + FP for each condition rating category are the sums achieved in the 3-fold cross validation, respectively. The precision and recall for each condition rating category are the averages achieved in the 3-fold cross validation, respectively. As a result, for each category, TP/(TP + FP) might not be exactly equal to the precision, and TP/(TP + FN) might not be exactly equal to the recall.

Table 8.4. The performance results of the proposed method and the baseline methods for predicting the condition ratings of the substructures.

CR [1]	TP + FN	Proposed method [2]				Baseline method #1 [3]				Baseline method #2 [4]			
		TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)	TP	TP + FP	P (%)	R (%)
"N"	54	54	56	96.5	100.0	54	56	96.5	100.0	54	56	96.5	100.0
"8"	66	55	65	84.8	83.3	56	66	85.2	84.8	51	78	65.4	77.3
"7"	1898	1836	1872	98.1	96.7	1799	1862	96.6	94.8	1780	1873	95.0	93.8
"6"	499	460	523	87.9	92.2	424	515	82.3	85.0	414	509	81.3	83.0
"5"	93	78	95	82.2	83.9	75	124	60.5	80.6	59	119	48.7	63.4
"4"	29	25	29	86.6	86.3	18	23	78.3	61.9	6	11	52.8	20.4
"3"	7	5	6	88.9	72.2	0	0	–	0.0	0	0	–	0.0
Macro-precision/macro-recall				89.3	87.8			83.2	72.4			73.3	62.5

[1] CR = condition rating category; "N" = "not applicable"; "8" = "very good condition"; "7" = "good condition"; "6" = satisfactory condition"; "5" = "fair condition"; "4" = "poor condition"; "3" = "serious condition".
[2] The proposed deep learning-based bridge deterioration prediction method, which uses the proposed RNN architecture (as per Fig, 1) for learning from the integrated data, the Isomap algorithm for data embedding, and the proposed multi-class focal loss function for addressing the imbalance in the data. It learns from the integrated bridge data from multiple sources.
[3] The first baseline method, which uses the RNN architecture, the Isomap algorithm, and the multi-class cross-entropy loss function (without addressing the imbalance). It learns from the integrated bridge data from multiple sources.
[4] The second baseline method, which uses the RNN architecture and the multi-class cross-entropy loss function (without addressing the imbalance). It only learns from the National Bridge Inventory (NBI) data.
[*] The other notations in the table: TP = true positives; FN = false negatives; FP = false positives; P = precision; R = recall; "–" = not applicable. The numbers of TP and TP + FP for each condition rating category are the sums achieved in the 3-fold cross validation, respectively. The precision and recall for each condition rating category are the averages achieved in the 3-fold cross validation, respectively. As a result, for each category, TP/(TP + FP) might not be exactly equal to the precision, and TP/(TP + FN) might not be exactly equal to the recall.

Table 8.5. The average performance results of the proposed method and the baseline methods.

Primary bridge components	Performance for predicting the condition ratings of the primary bridge components					
	Proposed method [1]		Baseline method #1 [2]		Baseline method #2 [3]	
	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
Decks	89.1	83.7	81.9	75.6	75.6	65.1
Superstructures	91.3	85.9	82.2	67.8	75.8	62.6
Substructures	89.3	87.8	83.2	72.4	73.3	62.5
Average	89.9	85.8	82.4	71.9	74.9	63.4

[1] The proposed deep learning-based bridge deterioration prediction method, which uses the proposed RNN architecture (as per Fig, 1) for learning from the integrated data, the Isomap algorithm for data embedding, and the proposed multi-class focal loss function for addressing the imbalance in the data. It learns from the integrated bridge data from multiple sources.
[2] The first baseline method, which uses the RNN architecture, the Isomap algorithm, and the multi-class cross-entropy loss function (without addressing the imbalance). It learns from the integrated bridge data from multiple sources.
[3] The second baseline method, which uses the RNN architecture and the multi-class cross-entropy loss function (without addressing the imbalance). It only learns from the National Bridge Inventory (NBI) data.
[*] The other notations in the table: P. = macro-precision; R. = macro-recall.

Table 8.6. The breakdown of the performance results of the proposed method for different bridge types.

Bridge type [1]	Number of bridges	Decks			Superstructures			Substructures		
		P (%) [2]	R (%)	STD	P (%)	R (%)	STD	P (%)	R (%)	STD
#1	1147	89.0	83.4	244.97	91.6	87.5	245.50	90.8	86.9	289.43
#2	1089	89.2	83.7	233.14	91.8	88.8	233.52	90.3	91.2	276.14
#3	317	86.8	85.7	68.36	92.5	83.5	68.82	93.5	94.0	80.64
#4	93	92.4	89.4	19.59	95.0	90.8	19.11	92.6	93.5	24.40

[1] #1 = prestressed concrete bridge; #2 = concrete bridge; #3 = steel bridge; #4 = timber bridge.
[2] P = macro-precision; R = macro-recall; STD = standard deviation (of the numbers of the bridges in different condition rating categories).

8.3.2 Performance of Predicting Element-Level Deficiency Quantities

The performance results of the proposed method in predicting the quantities of three common types of element-level deficiencies of the bridges – pier spall, deck delamination, and girder crack – are summarized in Table 8.7. The breakdown of the performance results for different bridge types are shown in Table 8.8. Two types of quantities for each type of deficiency were predicted: the total number of deficiency instances of the same type of deficiency (e.g., spall) on the same type of element (e.g., pier) of a bridge, and the total length (or area) of the deficiency instances. Only the performance results of learning from the integrated bridge data using the proposed method were reported for two main reasons. First, to the author's best knowledge, there is no existing data-driven prediction method that is able to predict the detailed quantity of a specific bridge element-level deficiency, which provides no benchmark for direct comparison. Second, learning from NBI

data solely is not applicable in this case, since they do not include such detailed data about bridge element-level deficiencies.

Table 8.7. The performance results of the proposed method for predicting the quantities of specific bridge element-level deficiencies.

Evaluation metric [1]	Performance for predicting the total number of deficiency instances of the element-level deficiencies			Performance for predicting the total length/area of deficiency instances of the element-level deficiencies		
	Pier spall	Deck delamination	Girder crack	Pier spall	Deck delamination	Girder crack
RMSE	0.4	0.2	0.3	2.1 (IN)	3.0 (SF)	2.0 (IN)
CV (%)	24.3	22.4	17.1	27.6	40.6	33.7
R^2	0.87	0.82	0.92	0.93	0.92	0.88

[1] RMSE = root mean square error; CV = coefficient of variation; R^2 = coefficient of determination.
[*] The other notations in the table: IN = inch and SF = square feet.

Table 8.8. The performance results of the proposed method for predicting the quantities of specific bridge element-level deficiencies for different bridge types.

Bridge type [1]	Number of bridges	Performance for predicting the total number of deficiency instances of the element-level deficiencies [2]											
		Pier spall				Deck delamination				Girder crack			
		RMSE	CV (%)	R^2	STD	RMSE	CV (%)	R^2	STD	RMSE	CV (%)	R^2	STD
#1	1147	0.4	25.6	0.85	2.03	0.2	17.9	0.79	0.11	0.2	9.8	0.94	0.73
#2	1089	0.4	25.1	0.87	1.35	0.3	33.6	0.53	0.17	0.4	37.8	0.87	2.18
#3	317	0.6	18.2	0.89	1.09	0.2	10.8	0.90	0.09	0.3	13.1	0.94	1.19
#4	93	0.4	12.8	0.91	0.70	–	–	–	–	–	–	–	–
Bridge type	Number of bridges	Performance for predicting the total length/area of deficiency instances of the element-level deficiencies [2]											
		Pier spall				Deck delamination				Girder crack			
		RMSE (IN)	CV (%)	R^2	STD	RMSE (SF)	CV (%)	R^2	STD	RMSE (IN)	CV (%)	R^2	STD
#1	1147	2.8	25.9	0.94	97.93	1.9	28.0	0.95	181.26	2.8	48.2	0.97	253.17
#2	1089	1.1	24.3	0.95	55.06	3	33.0	0.89	275.26	1.5	26.3	0.89	114.01
#3	317	2.9	32.4	0.90	115.25	4.5	59.7	0.94	751.1	1.1	17.1	0.88	55.27
#4	93	0.8	18.2	0.94	28.96	–	–	–	–	–	–	–	–

[1] #1 = prestressed concrete bridge; #2 = concrete bridge; #3 = steel bridge; #4 = timber bridge.
[2] RMSE = root mean square error; CV = coefficient of variation; R^2 = coefficient of determination; STD = standard deviation (of total lengths/areas or numbers of deficiency instances).
[*] The other notations in the table: IN = inch; SF = square feet; "–" = not applicable.

The performance results show that the proposed method performed well in predicting the quantities of the element-level deficiencies. When predicting the total number of deficiency instances of a specific element-level deficiency (e.g., pier spall of a specific bridge), it achieved an average

RMSE, CV, and R^2 of 0.3, 21.3%, and 0.87, respectively. This could indicate that, for a specific deficiency on a specific bridge element, the difference between the actual total number of the deficiency instances and the predicted total number is 0.3; the difference varies with respect to the mean of the total numbers of all the deficiencies of the same type by 21.3%; and 0.87 of the variance of the actual total number (with respect to the mean) can be correctly explained by the proposed method. When predicting the total length/area of deficiency instances of a specific element-level deficiency, it achieved an average RMSE, CV, and R^2 of 2.4 (inch or square feet), 34.0%, and 0.91. This could indicate that, for a specific deficiency on a specific element, the difference between the actual total length and the predicted total length is 2.4 inch; the difference varies with respect to the mean of the total lengths of all the deficiencies of the same type by 34.0%; and 0.91 of the variance of the actual total length can be correctly explained by the proposed method.

The proposed method achieved a comparatively higher CV, 34.0% vs. 21.3% (i.e., a lower level of performance), when predicting the total length/area, compared to predicting the total number. The higher CV might be caused by two main reasons. First, compared the total number (generally up to 10 in the dataset), the totals of the lengths/areas of the deficiencies span a much wider range (e.g., up to 2,000 inches in the dataset). But the mean of the lengths/areas is much smaller than those extreme length/area values, due to the rareness of such extremely-severe deficiencies. When comparing the errors of predicting the lengths/areas of such deficiencies to the mean for calculating CV, the value of CV got largely amplified by the smaller mean. Second, due to the complex mechanisms that affect the propagation of bridge deficiencies and the complexity of data reflecting these mechanisms, predicting the lengths/areas of the deficiencies is, naturally, much more challenging than predicting their total number. Same as the results for predicting the condition

ratings, the results for predicting the deficiency quantities (as per Table 8.8) also showed that data variability, not only data size, is essential for improved prediction performance.

8.3.3 Error Analysis

Two main types of errors that contributed to the incorrect predictions were identified. First, the errors that occurred during the information extraction and data integration steps could have propagated into deterioration prediction errors. Table 8.9 summarizes the precision and recall error rate of each step. The precision error rates increased from 4.3% for the information extraction to 12.2% for the data linking, and then decreased from 12.2% to 4.1% for the data fusion. The decrease was mainly because some of the incorrectly extracted and linked data/information were "corrected" in the fusion step. The recall error rates showed a constant increasing trend, increasing from 12.7% for the extraction to 16.8% for the fusion. The increase was mainly because some the data/information that should be extracted and integrated were incorrectly "dumped" and could not be recovered by the subsequent algorithms. For example, for the following sentence, the deficiency concept name was incorrectly extracted as "scattered transverse cracks" (i.e., the categorical quantity measure information entity "scattered" was incorrectly extracted as a deficiency information entity), but was fused into a correct name "transverse crack"; yet, the missed extraction of "scattered" as a categorical quantity measure was not recoverable: "The steel truss slabs have scattered leaching transverse cracks in the soffit." (WSDOT 2016).

Second, the dimensionality and the imbalance of the integrated bridge data showed negative impacts on the prediction performance. The Isomap algorithm was used to reduce the dimensionality of the bridge data for effective prediction model learning. This algorithm, like all other manifold learning algorithms, caused information loss during the embedding process. The information that was lost during the embedding could be indicative of bridge deterioration patterns.

In addition, although the multi-class focal loss function was able to address the data imbalance problem to a great extent, the imbalance problem was not fully resolved (and, indeed, cannot be fully resolved by any existing method). For example, as shown in Table 8.2, using the loss function was able to improve the precision from 61.0% to 79.3% and the recall from 63.6% to 74.2%, for the minority class of deck condition rating categories "5". But the improved precision and recall are still lower than the averages (precision = 89.1% and recall = 83.7%). The lost information and the incompletely-resolved imbalance posed challenges to the prediction algorithm and made it generate some prediction errors.

Table 8.9. Error rates of the main algorithms of the proposed bridge data analytics framework.

Component	Algorithm	Precision / error rate (%)	Recall / error rate (%)
Information extraction	Ontology-based semi-supervised conditional random field (CRF)-based information extraction algorithm	95.7 / 4.3	87.3 / 12.7
	Semantic neural network ensemble-based relation extraction algorithm	88.7 / 11.3	85.5 / 14.5
Data integration	Unsupervised data linking algorithm	87.7 / 12.2	84.9 / 15.1
	Hybrid data fusion algorithm [1]	95.9 / 4.1	83. 2 / 16.8
Data analytics	Deep learning-based bridge deterioration prediction algorithm	89.9 / 10.1	85.8 / 14.2

[1] The hybrid data fusion algorithm includes two main sub-algorithms: (1) named entity normalization algorithm for fusing the multiple concept names of the same entity and (2) numerical data fusion algorithm for fusing the multiple deficiency measures of the same type of deficiency. Because there is not ground truth for numerical data fusion, the precision and recall of the hybrid data fusion algorithm only represent those of the normalization sub-algorithm, not including the numerical data fusion sub-algorithm.

CHAPTER 9 – CONCLUSIONS, CONTRIBUTIONS, LIMITATIONS, AND RECOMMENDATIONS FOR FUTURE RESEARCH

9.1 Conclusions

9.1.1 Conclusions for the Proposed Bridge Deterioration Knowledge Ontology

In this thesis, a new ontology (BridgeOnto) for representing bridge deterioration knowledge for supporting semantic information and relation extraction from textual bridge inspection reports was developed. It captures and represents bridge deterioration knowledge in five primary aspects with in-depth classifications and rich multimodality views, including bridge element, deficiency, deficiency cause, maintenance action, and their related attributes. It aims to facilitate information and relation extraction from the reports based on content and domain-specific meaning. The ontology was verified through answering competency questions and automated consistency and redundancy checking, and was validated through human expert interviews and application-oriented validation. The verification results showed that the ontology was able to answer all the competency questions and passed the consistency and redundancy checks. The expert interview results indicated that the ontology is very representative, covering the main aspects of the bridge deterioration knowledge, clear, effective in classification, consistent, very concise, very easy to navigate, and extendable. The application-oriented validation results showed that the ontology was effective in supporting the information and relation extraction: (1) compared to without using the semantic features defined by the ontology, using such features in assessing the similarities between information entities improved the precision and recall of the information extraction by 50.5% and 45.2%, respectively (as shown in Section 4.3.2.1); and (2) compared to without using the semantic features, using such features for representing the configurations improved the configuration-based accuracy of the relation extraction by 7.3% (as shown in Section 5.3.2).

9.1.2 Conclusions for the Proposed Information Extraction Method and Algorithm

In this thesis, a new ontology-based, semi-supervised conditional random fields (CRF)-based information extraction (IE) method and algorithm for extracting information from textual bridge inspection reports was proposed and developed. The proposed IE algorithm allows for capturing the dependency structures as well as the distributions of a small set of fixed labeled data and a large set of unlabeled data semantically and simultaneously in a concave machine-learning function. It was hypothesized that, by dynamically adapting itself to unseen instances through further learning from a large collection of unlabeled data, the IE algorithm can achieve the goal of extracting information about existing deficiencies and performed maintenance actions from bridge inspection reports with reduced human effort, as well as high precision and recall performance. To test this hypothesis and to fine-tune the parameters of the IE algorithm, six primary experiments with controlled groups were conducted. The experimental results indicated that the algorithm achieved an average precision, recall, and F-1 measure of 94.1%, 87.7% and 90.7%, respectively, using only 175 human-annotated sentences from the I-35W Bridge 2006 inspection report as a fixed labeled dataset. The baseline, supervised CRF-based IE algorithm, achieved an average precision, recall, and F-1 measure of 85.8%, 80.8% and 83.1%, respectively, with 630 human-annotated sentences. The experiment results, thus, prove that the hypothesis is true. In addition, the following conclusions were drawn from the results: (1) the knowledge-based semantic similarity indicator provided an effective way for measuring token-to-token semantic similarities for improved IE performance; (2) the semantic similarity of the context in which a token appears affected the IE performance; (3) only using the most-similar entity class sequence for each unlabeled sentence achieved an optimal performance; (4) the optimal regularization item weight of the proposed IE algorithm was 0.4; and (5) the 2006 I-35W Mississippi River Bridge inspection

report provided a reliable source for creating a fixed labeled dataset, and the optimal size of human-annotated sentences from this report for creating such a dataset was 175.

9.1.3 Conclusions for the Proposed Relation Extraction Method and Algorithm

In this thesis, a new semantic neural network ensemble (NNE)-based dependency parsing method and algorithm for extracting dependency relations from textual bridge inspection reports was proposed and developed. The proposed parsing algorithm automatically links the isolated words into concepts and represents the semantically-low concepts in a semantically-rich structured way that is ready for bridge data analytics. A set of experiments were conducted to evaluate the performance of the parsing algorithm. The experimental results showed that the algorithm achieved an average semantic information element (SIE)-level precision, recall, and F-1 measure of 96.6%, 90.4%, and 93.3% with a margin of error of 3.8%, 4.4%, and 3.8%, and an semantic information set (SIS)-level precision, recall, and F-1 measure of 88.2%, 81.5%, and 84.7% with a margin of error of 5.4%, 5.8%, and 5.4%, respectively. The experimental results also showed that the semantic NNE-based algorithm was effective. First, the proposed semantic distributed feature representation improved the accuracy by 7.3%, compared to the representation without using the semantic features. Second, the proposed similarity-based sampling method improved the accuracy by 14.9%, compared to the method using cross-validation partitioning. Third, by taking an ensemble learning-based approach, the proposed algorithm improved the accuracy by 20.9%, on average, compared to the baselines using a single classifier.

9.1.4 Conclusions for the Proposed Data Linking Method and Algorithm

In this thesis, a new spectral clustering (SC)-based data linking method and algorithm for linking data extracted from textual bridge inspection reports was proposed and developed. The proposed

method offers a new concept similarity assessment method, a new sequential record similarity assessment method, and an improved SC method. A set of experiments were conducted to evaluate the performance of the data linking algorithm in linking the records extracted from the reports. The experimental results showed that the algorithm performed well: on average, it achieved a precision, recall, and F-1 measure of 96.2%, 88.3%, and 92.1%, respectively. In addition, five main conclusions were drawn from the results. First, different term similarity scoring functions showed similar performance, when the similarity assessment was conducted after stemming. Second, the similarities of the concepts in bridge inspection reports were better assessed by the similarities of their most-similar terms, rather than the similarities of all their terms. The impact of considering the relative positions of the terms on concept similarity assessment was insignificant. Third, a partitioning threshold value in the range of 0.05 to 0.15 was found optimal. Fourth, the sequential record similarity assessment method and the iterative bi-partitioning method were significantly effective. Fifth, the unsupervised pre-classification was not found effective in improving the performance of the data linking, because the sizes of the graphs were not large enough to benefit from size reduction. But, theoretically, pre-classification might show effectiveness in other cases/applications that deal with larger graph sizes.

9.1.5 Conclusions for the Proposed Data Fusion Method and Algorithm

In this thesis, a new hybrid data fusion method and algorithm for fusing data extracted from textual bridge inspection reports into a unified representation was proposed and developed. At the cornerstone of the proposed method are two algorithms for fusing concept names and numerical data, respectively: an unsupervised named entity normalization algorithm and an entropy-based numerical data fusion algorithm. A set of experiments were conducted to evaluate the performance of the two algorithms. Four main conclusions were drawn from the experimental results. First, the

concept ranking function with the corpus-statistic, term-position, and term-sequence scores, compared to those with the other combinations of the three scores, was more effective in fusing the concept names. Second, the concept selection rule using the part-of-speech (POS) patterns, compared to that without using the patterns, was more effective in fusing the names. Third, the numerical data fusion algorithm was stable, up to an uncertainty level of 2.0. Fourth, the proposed data fusion method was effective in fusing data extracted from the reports for supporting enhanced bridge deterioration prediction. Compared to learning from the unfused report data, learning from the report data fused by the proposed method improved the accuracy for predicting the condition ratings of the decks, superstructures, and substructures by 8.0%, 8.5%, and 7.9%, respectively.

9.1.6 Conclusions for the Proposed Data-Driven Bridge Deterioration Prediction Method and Algorithm

In this thesis, a new data-driven, deep learning-based prediction method and algorithm for predicting bridge deterioration was proposed and developed. It learns from integrated bridge data from multiple sources to predict the condition ratings of bridges and to predict the quantities of specific bridge element-level deficiencies. The proposed method includes three primary components: manifold learning, RNN modeling, and cost-sensitive learning. A set of experiments were conducted to evaluate the performance of the proposed algorithm. The experimental results showed that the proposed method achieved an average macro-precision and macro-recall of 89.9% and 85.8% when predicting the condition ratings of the primary bridge components (i.e., decks, superstructures, and substructures), and achieved an average RMSE, CV, and R^2 of 1.3, 27.6%, and 0.89 when predicting the quantities of three common types of bridge element-level deficiencies (i.e., pier spall, deck delamination, and girder crack), respectively. In addition, experiments were conducted to compare the performance of the proposed approach to the

performance of existing data-driven bridge deterioration prediction approaches, which mostly learn from bridge inventory data (mainly, the NBI data or similar inventory data collected by different counties) for predicting the deterioration. The comparison results showed that, when predicting the condition ratings, the proposed approach – by learning from the integrated bridge data from multiple sources – improved the precision by 15.0% and the recall by 23.3%. The experimental results indicate the promise of the proposed data-driven bridge deterioration prediction approach (the proposed bridge data analytics framework) in supporting enhanced data-driven bridge deterioration.

9.2 Contributions to the Body of Knowledge

9.2.1 Contributions of the Proposed Bridge Deterioration Knowledge Ontology

This research contributes to the body of knowledge by offering an important effort in bridge deterioration knowledge modeling. A domain-specific, formalized bridge deterioration knowledge ontology was proposed to capture bridge deterioration knowledge in five main aspects: bridge element, deficiency, deficiency cause, maintenance action, and their related attributes. The ontology advances the knowledge modeling efforts in the bridge domain by capturing the aforementioned bridge deterioration knowledge, with sufficient breadth, depth, classifications, and multimodality views. The ontology has shown effectiveness in adequately supporting semantic information and relation extraction from bridge inspection reports and, hence, is expected to be able to support similar text analytics tasks in the bridge domain.

9.2.2 Contributions of the Proposed Information Extraction Method and Algorithm

This research contributes to the body of knowledge in three primary ways. First, it offers a novel semantic computational method for semi-supervised CRF-based IE. The proposed IE algorithm

semantically and simultaneously captures the dependency structures as well as the distributions of a small set of fixed labeled data and a large set of unlabeled data, in a semi-supervised yet concave objective function for machine learning. Its capability of dynamically adapting itself to unseen instances by further learning from the unlabeled data and its concavity nature enable the needed IE to be conducted accurately and in an efficient way that requires less human effort. Second, unlike most of the existing IE efforts in the construction domain that focused on rule-based IE methods, the proposed IE algorithm takes a semi-supervised machine learning-based approach. This makes the proposed algorithm easily reusable and extendable for supporting other IE application needs in this particular domain, because no extraction rules need to be created or adapted. Third, and most importantly, the use of the proposed IE algorithm provides improved access to a large amount of information on bridge deficiencies and maintenance actions, which have typically been unexploited and buried in bridge inspection reports. The extracted information has shown effectiveness in improving the performance of bridge deterioration prediction, and could create new knowledge on existing bridge deficiencies and maintenance strategies, enhance the understanding of bridge deterioration, and result in enhanced maintenance decision making.

9.2.3 Contributions of the Proposed Relation Extraction Method and Algorithm

This research contributes to the body of knowledge in four primary ways. First, it offers a way of leveraging domain-specific semantics – as captured by the semantic features – for better supporting the analysis of highly technical, domain-specific text for improved extraction of dependency relations. Second, this research offers a new sampling method that utilizes similarities measured in multiple feature spaces as a collective criterion to sample data into meaningful clusters for better supporting ensemble learning. The proposed method allows for generating meaningful clusters that contain the densely- and sparsely-distributed as well as the correctly and incorrectly densely-

distributed data. Third, this research provides a novel parsing approach that is semantic, NN-based, and ensemble learning-based. It uses a set of constituent NN classifiers and a combiner SVM classifier to collectively capture the complex distributions of data instances. It was thus able to provide better parsing performance than that achieved by conventional dependency parsing methods, which only rely on a single classifier. Although the experimental results focused on dependency parsing, the applicability of the method is not limited to this case. Rather, it is a generic machine learning approach, which has the potential to support many other data-driven applications, such as text classification and sentiment analysis. When applying the proposed method to a different knowledge domain or application, one can choose to use another type of constituent or combiner classifier and test if the classifier of choice can improve the performance for the application at hand. Fourth, and most importantly, this research offers an automated method to extract word-to-word dependency relations from bridge inspection reports. It automatically links the isolated words into concepts and represents the unstructured and semantically-low concepts in a semantically-rich structured way that is ready to be used in data analytics for predicting bridge deterioration. The proposed method, therefore, allows the use of untapped wealth of data in the unstructured reports for bridge deterioration prediction.

9.2.4 Contributions of the Proposed Data Linking Method and Algorithm

This research contributes to the body of knowledge in four primary ways. First, it offers a new data linking method, which leverages improved spectral clustering to analyze the similarities between data instances for effectively linking data in an unsupervised way, without human involvement. The method offers opportunities for additional data analytics, as we are able to link data extracted from textual bridge inspection reports. This takes us one step closer towards the ability to learn from heterogeneous data – including unstructured data – for enhanced data-driven

bridge deterioration prediction and maintenance decision making. Second, it offers a new concept similarity assessment method, which does not need prerequisite contextual information or taxonomy-based mapping. The method provides an effective alternative way to assess concept similarity, when such prerequisites are not readily available. It also makes the similarity assessment free from dependence on external information and knowledge sources, whose quality could heavily affect the assessment performance. Third, it offers a new record similarity assessment method, which, unlike the commonly-used vector representations of attribute similarities, takes similarity assessment dependencies into consideration. Using dependencies to break down record-level similarity assessment into sequences of attribute-level tasks leads to reduced similarity-assessment complexities and reduced false positives. Fourth, it offers an improved SC method, which uses iterative bi-partitioning to automatically identify the optimal number of target clusters. This is important, because, otherwise, repeated clustering experiments are needed to manually identify the optimal number (either through trial-and-error or using the elbow method). In addition to largely reducing the human involvement in the manual process, compared to the elbow method, the iterative bi-partitioning was able to more accurately identify this number and improve the precision of clustering. The improved method, thus, extends the applicability of the original SC method to cases where high clustering performance is critical or where repeating the clustering experiments for each dataset in a large collection (e.g., each bridge inspection report) becomes heavily time-consuming or even practically impossible.

9.2.5 Contributions of the Proposed Data Fusion Method and Algorithm

This research contributes to the body of knowledge in two primary ways. First, it offers a new unsupervised named entity normalization algorithm for fusing concept names without human involvement. The algorithm captures both surface-form and abstraction-detailedness variations in

concept names to fuse them into canonical identifier names with balanced abstraction and detailedness. It, thus, extends the state of the art in named entity normalization, where most of the existing methods rely heavily on human-developed dictionaries/data and can only surface-form variations to fuse concept names into their canonical forms. Second, this research offers a new entropy-based data fusion algorithm. The algorithm uses data discretization to define the interval-based representation of the fused data, and leverages information entropy to fuse data that are complementary into a single representative representation. It, thus, adds to the state of the art in numerical data fusion, where most of the existing methods focus on fusing data that are conflicting and/or imprecise.

9.2.6 Contributions of the Proposed Data-Driven Bridge Deterioration Prediction Method and Algorithm

This research contributes to the body of knowledge in two primary ways. First, it offers a new computational method that is able to learn from highly dimensional and imbalanced bridge data for enhanced bridge deterioration prediction. Compared to existing methods which mostly leave such data challenges understudied or even untouched, the proposed bridge deterioration prediction method uses manifold learning to embed the high-dimensional data into a low-dimensional space and utilizes cost-sensitive learning to address the imbalance in the data. It, thus, offers new knowledge on how to effectively use bridge data that are very challenging in terms of dimensionality and imbalance for better predicting bridge deterioration. Second, and most importantly, it offers a novel data-driven bridge deterioration prediction approach. The proposed approach leverages advanced machine learning-based data analytics methods – semantic information extraction method, unsupervised data integration method, and deep learning-based prediction method – to allow for the extraction, integration, and analysis of multi-source

heterogeneous bridge data in an integrative manner for enhanced bridge deterioration prediction. On one hand, the proposed approach goes beyond the current state of the art in data analytics, where data in heterogeneous formats (i.e., structured and unstructured) are mostly analyzed separately. On the other hand, it goes beyond the current state of the art in data-driven bridge deterioration prediction, where existing methods mostly use abstract bridge inventory data to predict – at a limited performance level – the condition ratings of bridges. By using the integrated bridge data from multiple sources, especially the previously-untapped textual bridge inspection reports, the proposed approach allows for the prediction of bridge condition ratings with improved performance and the prediction of the quantities of element-level deficiencies. The use of the proposed bridge data analytics framework has the potential to transform the way decision makers in the bridge domain use and interact with the scattered and heterogeneous data – in an integrated and analyzed manner.

9.3 Limitations and Recommendations for Future Research

9.3.1 Limitations of the Proposed Bridge Deterioration Knowledge Ontology and Recommendations for Future Research

Two main limitations of this research are acknowledged. First, the developed bridge deterioration knowledge ontology was validated using a small number of expert interviews due to the challenges in recruiting qualified participants. A total of eight expert participants from both academia and industry were recruited for the interviews. Although these experts are very experienced in the bridge domain and can sufficiently validate the ontology, a larger number of qualified participants would allow for validating the ontology more extensively. Second, the ontology was implemented and validated in supporting information and relation extraction from bridge inspection reports. Although it was primarily designed to support this particular application, the ontology, like all

other ontologies, is meant to be reusable. Hence, the reusability of the ontology in other applications (e.g., using it for another text analytics application) was not evaluated.

Three main future research directions could be pursued to extend or improve this research. First, further validate the developed ontology using a larger number of qualified experts, in order to identify the aspects of the ontology that need further improvements and the ways of how it should be improved. Second, further validate the performance of the ontology in supporting other types of text analytics applications (e.g., text classification), in order to evaluate its reusability in these applications. Third, maintain the ontology current in its representation and representativeness of its domain by adding, reclassifying, and/or modifying its concepts and relations.

9.3.2 Limitations of the Proposed Information Extraction Method and Recommendations for Future Research

Two main limitations of this research are acknowledged. First, the proposed IE method only considers a token's context defined by its preceding and succeeding tokens (i.e., context window of size one). A larger window size (e.g., size of two) could capture the needed context when dealing with noise in text (e.g., "of possible" is the noise in "drilling of possible stress relief holes") for a better IE performance. Second, like any other ontology-based method, the performance of the proposed IE method partially depends on the coverage and quality of ontology used. As shown from the error analysis (Section 4.3.2.2), the coverage and token-level ambiguity of the ontology affected the IE performance.

Three main future research directions could be pursued to extend or improve this research. First, in future applications of the proposed IE method, model a larger context window and test if it can better deal with the noise caused by a small context window. This can be achieved by representing each current token with, in addition to its own features, features defined by the two (i.e., in the

case of using context window size of two) closest tokens to the left and right of the token. Second, evaluate the impacts of using different ontologies – which could naturally vary in coverage, structure, semantics, terminology, etc. – on the performance of information extraction. Third, extend the proposed IE method in a way that it can learn from a large amount of domain-general labeled text that is readily available to extract information from domain-specific text. If successful, the extended IE method would significantly save human-annotation efforts and, thus, benefit IE applications in various domains, not only information extraction from inspection reports in the bridge domain.

9.3.3 Limitations of the Proposed Relation Extraction Method and Recommendations for Future Research

Three main limitations of this research are acknowledged. First, the error analysis (in Section 5.3.4) has shown that the errors in the POS tags have negatively affected the performance of the proposed dependency parsing method. One main reason for the POS tagging errors is that the NLTK POS tagger, like all other taggers, was trained using general-domain text [e.g., the Wall Street Journal (WSJ) dataset]. Second, in this research, three types of SIE-to-SIE dependency relations were defined to support the representation of the text. These relations were chosen because they are representative of the information needed for better predicting bridge deterioration, yet they are not too abundant or complex to the extent of causing extra errors in the extraction. The error analysis, however, revealed that they are sometimes not enough to capture all the needed information. Third, the proposed method is limited in dealing with the imbalance in the transition types/classes. As a result, the precision and recall of the majority class (i.e., “shift”) were higher than those of the minority classes (i.e., “left-arc” and “right-arc”). Thus, for the confusion matrices (Figure 5.9), the precision and recall of each individual class and the average accuracy should be interpreted jointly.

Four main future research directions could be pursued to extend or improve this research. First, investigate the use of different types of neural network architectures (e.g., LSTM) and different types of transition-based approaches (e.g., top-down and bottom-up predictions) for supporting dependency relation extraction. Second, develop a domain-specific POS tagger and test its impact on the performance of dependency relation extraction. Third, in the case of using the proposed method for extracting dependency relations from bridge inspection reports, explore the use of additional SIE-to-SIE dependency relations (e.g., ET-DC-SM relations) to identify the optimal number and types of dependency relations. In the case of using it for other types of text, study the characteristics of the text to identify the optimal number and types. Fourth, explore the use of data sampling methods (e.g., random over-sampling method and synthetic minority over-sampling technique), in order to balance the number of configurations in different transition classes for further improving the performance of dependency relation extraction.

9.3.4 Limitations of the Proposed Data Linking Method and Recommendations for Future Research

Two main limitations of this research are acknowledged. First, the concept similarity assessment method is string-based and is, thus, limited in assessing the similarities of synonyms (e.g., "battledeck" and "orthotropic steel deck plate") and acronyms (e.g., "delam" and "delamination"). If such concepts are frequent in a particular knowledge domain, the method will require adaptation (e.g., incorporating a gazetteer list containing the commonly-used synonyms and acronyms in that domain) prior to adoption in the new domain. Second, the dependencies used in the record similarity assessment method are specific to the domain of knowledge covered in the bridge reports. Because the types of dependencies naturally vary from one knowledge domain to another, the

defined dependencies may need to be adapted (through modifications and/or extensions), if the method is used in a different domain.

Three main future research directions could be pursued to extend or improve this research. First, extend the SC-based data linking method to allow for efficient updating of the linking results when changes in the datasets occur (e.g., addition and/or deletion of one or more records). For example, only updating the linking results, without re-doing the entire linking, could be computationally-efficient. Second, study how to learn word embeddings from domain-specific text corpora in an unsupervised way and use the learned embeddings for better assessment of concept similarities. Third, study how to use sequential deep neural networks (e.g., recurrent neural networks) to automatically capture dependencies among the attribute similarity assessments for improved linking performance.

9.3.5 Limitations of the Proposed Data Fusion Method and Recommendations for Future Research

Three main limitations of this research are acknowledged. First, the unsupervised named entity normalization algorithm uses the normalized Google distance to assess the associations of words for selecting identifier concept names. Half of the normalization errors were caused by the incorrectly-assessed associations, where the total error rate of the algorithm is 5.6%. This distance was mainly developed for assessing the associations of words in general-domain text. Second, the entropy-based data fusion algorithm focuses on fusing data from a single type of source (e.g., deficiency measures from the text). It needs modifications and/or extensions when used for fusing multi-modal data (e.g., deficiency measures from text, images, and sensors). Third, the bridge deterioration prediction models were developed using the decision tree algorithm. Although this algorithm is suitable for the validation purpose, it is limited in dealing with highly-dimensional

and imbalanced data such as bridge data and could be, thus, limited in showing the significance of learning from fused report data over learning form unfused report data.

Three main future research directions could be pursued to extend or improve this research. First, develop a word-association assessment measure that can better adapt to domain-specific text and test its impact on named entity normalization. Second, extend the proposed entropy-based fusion algorithm to a multi-level context-based fusion algorithm to further capture the context of data (e.g., sensing devices and their reliability) for fusing data that are complementary and multi-modal at the same time. Third, further evaluate the performance of the proposed hybrid data fusion method in fusing data extracted from textual inspection reports for supporting bridge deterioration prediction. It is expected that, compared to learning from unfused report data, learning from fused report data is able to show an even more significant improvement in the prediction performance, if a machine learning algorithm that is better than the decision tree algorithm in dealing with data dimensionality and imbalance is used.

9.3.6 Limitations of the Proposed Data-Driven Bridge Deterioration Prediction Method and Recommendations for Future Research

Five main limitations of this research are acknowledged. First, the performance of the proposed bridge deterioration prediction method was evaluated in predicting three types of bridge element-level deficiencies. Second, the proposed method uses a unidirectional RNN architecture. Although the experimental results showed the effectiveness of this architecture, a bidirectional RNN architecture would allow for capturing both forward and backward information about the sequential changes of bridge conditions (e.g., the information about why the condition of a bridge in the current year is improved compared to that in the previous year). It could, thus, potentially help improve the prediction performance. Third, the proposed method predicts one type of target

class (e.g., the condition ratings of decks) at a time. Fourth, the proposed method is limited in analyzing the time-series patterns of bridge deterioration (e.g., how temperature changes at different temporal granularities affect the deterioration). Fifth, the proposed method primarily focuses on bridge deterioration prediction. It is limited in understanding the deterioration of bridges and its impacts on bridge performance and maintenance decision making.

Four main future research directions could be pursued to improve the proposed method. First, further test the performance of the proposed method in predicting other types of deficiencies. Although some variability in the performance may occur, a similar performance is expected if the data characteristics are similar. Second, develop a bidirectional RNN architecture and test its impact on the performance and the computational efficiency of data-driven bridge deterioration prediction. Third, incorporate transfer learning techniques into the proposed method to allow for the use of only one single base model for capturing the underlying distributions of the input data and for the adaptations to different types of target classes with minimum computational resources. Fourth, incorporate time-series analysis techniques into the proposed method to allow for considering the temporal patterns of deterioration-related factors (e.g., temperature) during the prediction, and test if the use of such patterns could improve the performance of the prediction.

Two main future research directions could be pursued to extend or improve this research, at the application level. First, deploy the proposed data analytics framework in the bridge management process of bridge agencies (e.g., state Departments of Transportation) to investigate the usability of the proposed framework in practice. This would allow for soliciting feedback from the target users of the framework (e.g., maintenance decision makers), in order to identify the pain points that might hinder direct adoption of the framework in the current practice and, accordingly, lay out of roadmap for promoting its full deployment. Second, apply the proposed framework to support

the deterioration prediction for other types of infrastructure (e.g., highway and dam) to evaluate its generalizability. The evaluation would provide insights on which aspects of the proposed data analytics methods are the most important in affecting their successful adaptations to other domains, thereby offering new general knowledge on how to better design and adapt existing analytics methods to other applications.

One critical future research direction that needs to be pursed in order to allow this research to bring more practical and broader impacts to the society is data-driven bridge maintenance decision making. Making well-informed maintenance decisions is rather complex, which requires considering many interrelated factors – not only the future conditions of the bridges, but also the probabilities of bridge failures, the cost-effectiveness of various maintenance strategies, and the impacts of bridge deterioration (including failures) and maintenance on the performance of the transportation network. Thus, in order to enhance our bridge maintenance decisions, further research is needed in two main directions: (1) study how to predict the time-dependent probabilities of bridge failures and the life-cycle cost-effectiveness of different maintenance scenarios, given the observed and predicted bridge conditions. To support the prediction, data from a number of important sources (in addition to those used in this research) should be exploited, including design and construction data from bridge information modeling, maintenance/accident data from textual bridge reports, and condition data from health monitoring sensors and inspection images; and (2) investigate how to automatically analyze and learn the impacts of the conditions, maintenance, and failures of the bridges on the safety, serviceability, and functionality of the entire transportation system. Transportation network data (e.g., traffic data, travel behavior data, and crash data) and socio-economic data (e.g., data about user travel cost and time, local economic conditions, and local accessibility) should be analyzed in integration with the data about bridge

deterioration and maintenance to capture the different factors that impact the safety, severability, and functionality of the overall system. These research efforts would create new knowledge on bridge deterioration and maintenance through the integrative analysis of the multi-source heterogeneous data, and pave the way for enabling safer, efficient, and cost-effective maintenance of our bridges.

REFERENCES

AASHTO. (2007). "AASHTO maintenance manual for roadways and bridges." American Association of State Highway and Transportation Officials (AASHTO), Washington, D.C.

AASHTO. (2009). "AASHTO transportation glossary." American Association of State Highway and Transportation Officials (AASHTO), Washington, D.C.

AASHTO. (2010). "AASHTO bridge element inspection guide manual." American Association of State Highway and Transportation Officials (AASHTO), Washington, D.C.

AASHTO. (2011). "Framework for a national database system for maintenance actions on highway bridges." American Association of State Highway and Transportation Officials (AASHTO), Washington, D.C.

Abdelkader, E.M., Zayed, T., and Marzouk, M. (2019). "Modelling the deterioration of bridge decks based on semi-Markov decision process." *International Journal of Strategic Decision Sciences*, 10(1):23-45.

Abdi, H. (2010). "Coefficient of variation." *Encyclopedia of Research Design*, 1(2010):169-171.

Abuzir, Y., and Abuzir, M. (2002). "Constructing the civil engineering thesaurus (CET) using ThesWB." *Proc., Intl. Workshop on Information Technology in Civil Engineering*, ASCE, Reston, VA, 400-402.

Ailon, N., Charikar, M., and Newman, A. (2008). "Aggregating inconsistent information: ranking and clustering." *Journal of the ACM*, 55(5):23.

Akhtar, M.S., Sikdar, U.K., and Ekbal, A. (2015). "IITP: Hybrid approach for text normalization in Twitter." *Proc., ACL 2015 Workshop on Noisy User-generated Text*, Association for Computational Linguistics, Stroudsburg, PA, 106-110.

Al Qady, M., and Kandil, A. (2010). "Concept relation extraction from construction documents using natural language processing." *Journal of Construction Engineering and Management*, 136(3):294-302.

Alberti, C., Weiss, D., and Petrov, S. (2015). “Improved transition-based parsing and tagging with neural networks.” *Proc., 2015 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 354-1359.

Albrecht, A., and Naumann, F. (2008). “Managing ETL processes.” *Proc., 34th Intl. Conf. on Very Large Data Bases*, Auckland, New Zealand, 12-15.

Alfonseca, E., and Manandhar, S. (2002). “An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery.” Proceedings of the1st International Conference on General WordNet, Mysore, India, 34-43.

Alhelbawy, A., and Gaizauskas, R. (2014). “Graph ranking for collective named entity disambiguation.” *Proc., 52nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 75-80.

Ananthakrishna, R., Chaudhuri, S., and Ganti, V. (2002). “Eliminating fuzzy duplicates in data warehouses.” *Proc., 28th Intl. Conf. on Very Large Data Bases*, VLDB, Hong Kong, China, 586-597.

APA. (2016). “Product guide: HDO/MDO plywood.” The Engineered Wood Association (APA), Tacoma, WA.

Appelt, D., Hobbs, J., Bear, J., Israel, D., and Tyson, M. (1993). “FASTUS: A finite-state processor for information extraction from real-world text.” *Proc. Intl. Joint Conf. on Artificial Intelligence*, Chambéry, France, 1172-1178.

ASCE. (2001). “Types and causes of defects and deterioration.” American Society of Civil Engineers (ASCE), <http://ascelibrary.org/doi/abs/10.1061/9780784405451.apb> (Sep. 20, 2015).

ASCE. (2013). “2013 report card for America’s infrastructure.” American Society of Civil Engineers (ASCE), <http://www.infrastructurereportcard.org/> (Jun. 25, 2016).

ASCE. (2017). “2017 report card for America’s infrastructure.” American Society of Civil Engineers (ASCE), < https://www.infrastructurereportcard.org/> (Jun. 25, 2017).

Attardi, G., and Dell'Orletta, F. (2009). "Reverse revision and linear tree combination for dependency parsing." *Proc. Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 261-264.

Babbar, R., and Schölkopf, B. (2017). "DiSMEC: Distributed sparse machines for extreme multi-label classification." *Proc., 10th ACM Intl. Conf. on Web Search and Data Mining*, Association for Computing Machinery, New York, NY, 721-729.

Bansal, M., Gimpel, K., and Livescu, K. (2014). "Tailoring continuous word representations for dependency parsing." *Proc., 52nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 809-814.

Bartolini, I., Ciaccia, P., and Patellam, M. (2002). "String matching with metric trees using an approximate distance." *Proc., 9th Intl. Symposium on String Processing and Information Retrieval*, Lisbon, Portugal, 271-283.

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). "A study of the behavior of several methods for balancing machine learning training data." *ACM SIGKDD Explorations Newsletter*, 6(1):20-29.

Belkin, M., and Niyogi, P. (2002). "Laplacian eigenmaps and spectral techniques for embedding and clustering." *Advances in Neural Information Processing Systems*, 585-591.

Ben-Hur, A., and Weston, J. (2010). "A user's guide to support vector machines." Data Mining Techniques for the Life Sciences. Methods in Molecular Biology (Methods and Protocols), Humana Press, Totowa, NJ.

Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B., and Rätsch, G. (2008). "Support vector machines and kernels for computational biology." *PLoS Computational Biology*, 4(10):e1000173.

Bergroth, L., Hakonen, H., and Raita, T. (2000). "A Survey of longest common subsequence algorithms." *Proc., 7th Intl. Symposium on String Processing and Information Retrieval*, Curuna, Spain, 39-48.

Bickel, S., Brückner, M., and Scheffer, T. (2007). "Discriminative learning for differing training and test distributions." *Proc., 24th Intl. Conf. on Machine Learning*, Association for Computing Machinery, New York, NY, 81-88.

Bień, J., Jakubowski, K., Kamiński, T., Kmita, J., and Kmita, P. (2007). "Railway bridge defects and degradation mechanisms." *Proc., Conf. on Sustainable Bridges: Assessment for Future Traffic Demands and Longer Lives*, Wroclaw, Poland, 105-116.

Bijen, J. (2003). *Durability of engineering structures: design, repair and maintenance*, Woodhead Publishing Limited, Cambridge, UK.

Bikel, D., Miller, S., and Schwartz, R. (1997). "Nymble: A high-performance learning name-finder." *Proc., 5th Conf. on Applied Natural Language Processing*, Washington, D.C., 194-201.

Bilenko, M., and Mooney, R.J. (2003). "Adaptive duplicate detection using learnable string similarity measures." *Proc., 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, 39-48.

Bilenko, M., Basil, S., and Sahami, M. (2005). "Adaptive product normalization: Using online learning for record linkage in comparison shopping." *Proc., 5th IEEE International Conference on Data Mining*, IEEE, Piscataway, NJ.

Bird S., Klein, E., and Loper, E. (2009). "Natural language processing with Python: Analyzing text with the natural language toolkit." O'Reilly Media, Inc.

Bishop, C.M. (2006). Pattern Recognition. Machine Learning, Springer Science & Business Media.

Blei, D., Ng, A., and Jordan, M. (2003). "Latent dirichlet allocation." *Journal of Machine Learning Research*, 3:993-1022.

Borthwick, A., Sterling, J., and Agichtein, E. (1998). "Exploiting diverse knowledge sources via maximum entropy in named entity recognition." *Proc., 6th Workshop on Very Large Corpora*, Montreal, Canada, 182.

Boström, H., Andler, S.F., Brohede, M., Johansson, R., Karlsson, A., Van Laere, J., Niklasson, L., Nilsson, M., Persson, A., and Ziemke, T. (2007). "On the definition of information fusion as a field of research." Technical Report, *HS-IKI-TR-07-006*, University of Skövde, Skövde, Sweden.

Bouchard, G. (2007). "Efficient bounds for the softmax function and applications to approximate inference in hybrid models." *Proc., 2007 Workshop for Approximate Bayesian Inference in Continuous/Hybrid Systems*, NIPS, San Diego, CA.

Boukhdhir, A., Lachiheb, O., and Gouider, M.S. (2015). "An improved MapReduce design of K-means for clustering very large datasets." *Proc., 12th Intl. Conf. of Computer Systems and Applications (*AICCSA), Marrakech, Morocco, 1-6.

Breiman, L. (1996). "Bagging predictors." *Machine Learning*, 24(2):123;140.

Brown, M.C., Gomez, J.P., Hammer, M. L., and Hooks, J.M. (2014). "LTBP bridge performance primer." *No. FHWA-HRT-14-052*, Federal Highway Administration (FHWA), Washington, D.C.

Brook, E.L., Rosman, D.L., and Holman, C.J. (2008). "Public good through data linkage: measuring research outputs from the western Australian data linkage system." *Australian and New Zealand Journal of Public Health*, 32(1):19-23.

Brookmeyer, R., and Crowley, J. (1982). "A confidence interval for the median survival time." *Biometrics*, 38(1):29-41.

Bu, G., Lee, J., Guan, H., Blumenstein, M., and Loo, Y.-C. (2014). "Development of an integrated method for probabilistic bridge-deterioration modeling." *Journal of Performance of Constructed Facilities*, 28(2):330-340.

Buchholz, S., and Marsi, E. (2006). "CoNLL-X shared task on multilingual dependency parsing." *Proc., 10th Conf. on Computational Natural Language Learning*, Association for Computational Linguistics Stroudsburg, PA, 149-164.

BuildingSMART. (2014). “IFC-Bridge.” <http://iug.buildingsmart.org/resources/itm-and-iug-meetings-2013-munich/infra-room/ifc-bridge-ifc-for-roads/view> (Apr. 25, 2016).

Bulskov, H., Knappe, R., and Andreasen., T. (2002). “On measuring similarity for conceptual querying.” *Flexible Query Answering Systems*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2522):100-111.

Caltrans. (2012). “Bridge inspection reports: Capitola Crossing Deck Truss.” California Department of Transportation (Caltrans), Sacramento, CA.

CDOT. (2009). “Structure No. 04601 North Bear Hill Rd. over Natchaug River Chaplin Indepth Inspection.” Connecticut Department of Transportation (CDOT), Newington, CT.

Cerullo, D., Sennah, K., Azimi, H., Lam, C., Fam, A. and Tharmabala, B. (2013). “Experimental study on full-scale pretensioned bridge girder damaged by vehicle impact and repaired with fiber-reinforced polymer technology.” *Journal of Composites for Construction*, 17(5):662-672.

Chai, T., and Draxler, R.R. (2014). “Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature.” *Geoscientific Model Development*, 7(3):1247-1250.

Chan, P.K., Schlag, M.D.F., and Zien, J.Y. (1994). “Spectral k-way ratio-cut partitioning and clustering.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1088-1096.

Chang, Chih-Chung, and Chih-Jen Lin. (2011). “LIBSVM: A library for support vector machines.” *ACM Transactions on Intelligent Systems and Technology*, 3(2011):27.

Chang, M., Maguire, M., and Sun, Y. (2018). “Stochastic modeling of bridge deterioration using classification tree and logistic regression.” *Journal of Infrastructure Systems*, 25(1): 04018041.

Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. (2002). "SMOTE: Synthetic minority over-sampling technique." *Journal of Artificial Intelligence Research*, 16:321-357.

Che, C., Xiao, C., Liang, J., Jin, B., Zho, J., and Wang, F. (2017). "An RNN architecture with dynamic temporal matching for personalized predictions of Parkinson's disease." *Proc., 2017 SIAM Intl. Conf. on Data Mining*, SIAM, Philadelphia, PA, 198-206.

Chen, D., and Manning, C. (2014). "A fast and accurate dependency parser using neural networks." *Proc., 2014 Conf. on Empirical Methods on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 740-750.

Chen, K., and Wang, S. (2011). "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):129-143.

Chen, W., and Zhang, M. (2015). "Dependency parsing models." *Semi-Supervised Dependency Parsing*, 1st ed., Springer, Singapore.

Chen, Z. (2016). "Lifelong machine learning for topic modeling and classification." University of Illinois at Chicago.

Cheng, H., Fang, H., He, X., Gao, J., and Deng, L. (2016). "Bi-directional attention with agreement for dependency parsing." *Proc., 2016 Conf. on Empirical Methods on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA.

Chiu, J.P., and Nichols, E. (2016). "Named entity recognition with bidirectional LSTM-CNNs." *Transactions of the Association for Computational Linguistics*, 4:357-370.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078.

Choi, J.D, and McCallum, A. (2013). "Transition-based dependency parsing with selectional branching." *Proc., 51st Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics Stroudsburg, PA, 1052-1062.

Chollet, F. (2015). Keras. <https://github.com/fchollet/keras> (June 21, 2019).

Christen, P. (2008) "Automatic Record linkage using seeded nearest neighbour and support vector machine classification." *Proc. 14th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, ACM, New York, NY, 151-159.

Christen, P. (2008), FEBRL: A Freely Available Record Linkage System with a Graphical User Interface. <http://users.cecs.anu.edu.au/~Peter.Christen/Febrl/febrl-0.3/febrldoc-0.3/manual.html> (Jun. 6, 2017).

Christen, P. (2012). *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection*, Springer Science & Business Media.

Cilibrasi, R., and Vitányi, P.M.B. (2005). "Clustering by compression." *IEEE Transactions on Information Theory*, 51(4):1523-1545.

Cilibrasi, R.L., and Vitanyi, P.M. (2007). "The google similarity distance." *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370-383.

Cochinwala, M., Kurien, V., Lalk, G., and Shasha, D. (2001) "Efficient data reconciliation." *Information Sciences*, 137 (1-4):1-15.

Collins, M. (2002). "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms." *Proc., ACL-02 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 1-8.

Collins, M. (2003). "Head-driven statistical models for natural language parsing." *Computational Linguistics*, 29(4):589-637.

Contreras-Nieto, C., Lewis, P., and Shan, Y. (2016). "Developing predictive models of superstructure ratings for steel and prestressed concrete bridges." *Proc., 2016 Construction Research Congress*, ASCE, Reston, VA, 859-868.

Cook, W., Barr, P.J., and Halling, M.W. (2013). "Bridge failure rate analysis." *Proc., Transportation Research Board 92nd Annual Meeting*, TRB, Washington D.C., 13-27.

Cormen, T., Leiserson, H., Charles E., and Rivest, R.L. (1990). "The Floyd-Warshall algorithm." *Introduction to Algorithms*, 1st ed. MIT Press and McGraw-Hill, Cambridge, MA.

Corro, L. Del, and Gemulla, R. (2013). "Clausie: Clause-based open information extraction." *Proc., 22nd Intl. Conf. on World Wide Web*, Rio de Janeiro, Brazil, 355-366.

Costa, J. A., and Hero, A.O. (2004). "Geodesic entropic graphs for dimension and entropy estimation in manifold learning." *IEEE Transactions on Signal Processing*, 52(8):2210-2221.

Cox, T.F., and Cox, M.A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.

Creary, P.A., and Fang, F.C. (2014). "Forecasting long-term bridge deterioration conditions using artificial intelligence techniques." I*nternational Journal of Intelligent Systems Technologies and Appli*cations, 13(4):280-293.

Cristani, M., and Cuel, R. (2005). "A survey on ontology creation methodologies*." International Journal on Semantic Web and Information Systems*, 1(2):49-69.

De Boer, P. T., Kroese, D. P., Mannor, S, and Rubinstein, R. Y. (2005). "A tutorial on the cross-entropy method." *Annals of Operations Research*, 134(1):19-67.

De Leone, R., and Minnetti, V. (2015). "Electre tri-machine learning approach to the record linkage problem." arXiv preprint, arXiv:1505.06614.

Destercke, S., Dubois, D., and Chojnacki, E. (2008). "Possibilistic information fusion using maximal coherent subsets." *IEEE Transactions on Fuzzy Systems*, 17(1):79-92.

Dey, D., Sarkar, S., and De, P. (1998). "Entity matching in heterogeneous databases: A distance-based decision model." *Proc., 31st Hawaii Intl. Conf. on System Sciences*, Kohala Coast, HI, 305-313.

Dietterich, T.G. (2000). "Ensemble methods in machine learning." *Multiple classifier systems*, Springer, Berlin, Heidelberg.

Doyle, S., Agner, S., Madabhushi, A., Feldman, M. and Tomaszewski, J. (2008). "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features." *Proc. 5th IEEE Intl. Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, Piscataway, NJ, 496-499.

Dozat, T., and Manning, C.D. (2017). "Deep biaffine attention for neural dependency parsing." *Proc., 5th Intl. Conf. on Learning Representations*, Ithaca, New York, 1-8.

Dozat, T., and Manning, C.D. (2018). "Simpler but more accurate semantic dependency parsing." *Proc., 5th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA.

Doyle, S., Agner, S., Madabhushi, A., Feldman, M., and Tomaszewski, J. (2008). "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features." *Proc., 5th IEEE International Symposium on In Biomedical Imaging: From Nano to Macro*, IEEE, Piscataway, NJ, 496-499.

Dyer, C., Ballesteros, M., Ling, W., and Matthews, A. (2015). "Transition-based dependency parsing with stack long short-term memory." *Proc., 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Intl. Joint Conf. on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 334-343.

Eden, J.A. (2018). "Development of a condition-based deterioration model for bridges in Rhode Island." University of Rhode Island, Kingston, RI.

Eisner, J.M. (1996). "Three new probabilistic models for dependency parsing: An exploration." *Proc., 16th Conf. on Computational Linguistic*, Copenhagen, Copenhagen, Denmark, 340-345.

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

El-Diraby, T. E., and Osman, H. (2011). "A domain ontology for construction concepts in urban infrastructure products." *Automation in Construction,* 20(8):1120-1132.

Elfeky, M.G., Verykios, V.S., and Elmagarmid, A.K. (2002). "TAILOR: A record linkage toolbox." *Proc., 18th Intl. Conf. on Data Engineering,* San Jose, CA, 17-28.

El-Gohary, N.M., and El-Diraby, T.E. (2010). "Domain ontology for processes in infrastructure and construction." *Journal of Construction Engineering and Management,* 136(7):730-744.

Elmagarmid, A.K., Ipeirotis, P.G., and Verykios, V.S. (2007). "Duplicate record detection: A survey." *IEEE Transactions on Knowledge and Data Engineering,* 19(1):1-16.

Elsebai, A., Meziane, F., and Belkredim, F. (2009). "A rule based persons names Arabic extraction system." *Communications of the IBIMA,* 11(6):53-59.

Elsner, M., and Charniak, E. (2008) "You talking to me? A corpus and algorithm for conversation disentanglement." *Proc., 2008 Annual Meeting of the Association for Computational Linguistics with the Human Language Technology Conf.,* Association for Computational Linguistics, Stroudsburg, PA, 834-842.

Elsner, Micha, and Schudy, W. (2000). "Bounding and comparing methods for correlation clustering beyond ILP." *Proc., Workshop on Integer Linear Programming for Natural Language Processing,* Boulder, CO, 19-27

Elworthy, D. (2000). "A finite state parser with dependency structure output." *Proc., Intl. Workshop on Parsing Technologies,* Trento, Italy.

Etzioni, O., Cafarella, M., Downey, D., and Popescu, A. (2005). "Unsupervised named-entity extraction from the Web: An experimental study." *Artificial Intelligence,* 165(1):91-134.

Fader, A., Soderland, S., and Etzioni, O. (2011). "Identifying relations for open information extraction." *Proc., Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, United Kingdom, 1535-1545.

Fakhraei, S., and Ambite, J.L. (2018). "NSEEN: Neural semantic embedding for entity normalization." arXiv preprint, arXiv:1811.07514.

Fan, M., Qiao, H., Zhang, B., and Zhang, X. (2012). "Isometric multi-manifold learning for feature extraction." *Proc., 2012 IEEE 12nd Intl. Conf. on Data Mining*, IEEE, Piscataway, NJ, 241-250.

Fang, Y., and Sun, L. (2018). "A Weibull distribution based semi-Markov process model for urban bridge deterioration prediction." *Proc., 97th Annual Meeting of Transportation Research Board*, Washington D.C., 18-04963.

Farhey, D. N. (2014). "Operational structural performances of bridge materials by deterioration trends." *Journal of Performance of Constructed Facilities*, 28(1):168-177.

Fellegi, I.P., and Sunter, A.B. (1969). "A theory for record linkage." *Journal of the American Statistical Association*, 64 (328):1183-1210.

Fensel, D., Van Harmelen, F., Klein, M., Akkermans, H., Broekstra, J., Fluit, C., Van Der Meer, J., Schnurr, H. P., Studer, R., and Hughes, J. (2000). "On-to-knowledge: Ontology-based tools for knowledge management." *Proceedings of the eBusiness and eWork*, 18-20.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). "METHONTOLOGY: From ontological art towards ontological engineering." *AAAI-97 Spring Symposium Series*, 33-40.

FHWA. (1995). "Recoding and coding guide for the structure inventory and appraisal of the nation's bridges." *FHWA-PD-96-01*, Federal Highway Administration (FHWA), Washington, D.C.

FHWA. (2001). "Reliability of Visual Inspection for Highway Bridges, Volume I: Final Report." Federal Highway Administration (FHWA), Washington, D.C.

FHWA. (2012). “Bridge inspector’s reference manual.” Federal Highway Administration (FHWA), Washington, D.C.

FHWA. (2015). “Deficient bridges by highway system 2014.” Federal Highway Administration (FHWA), Washington, D.C., <https://www.fhwa.dot.gov/bridge/nbi/no10/defbr14.cfm#a> (Sep. 20, 2015).

FHWA. (2016). “The LTBP deterioration modeling algorithm - An innovative approach to accurate deterioration modeling.” Federal Highway Administration (FHWA), Washington, D.C., <https://www.fhwa.dot.gov/publications/ltbpnews/15073.cfm> (Dec. 4, 2017).

FHWA. (2019). “National bridge inventory.” Federal Highway Administration (FHWA), Washington, D.C., < https://www.fhwa.dot.gov/bridge/nbi/ascii.cfm> (June 21, 2019).

Fisher, J., Christen, P., Wang, Q., and Rahm, E. (2015). “A clustering-based framework to control block sizes for entity resolution.” *Proc., 21st ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, 279-288.

Forney, G. (1973). “The Viterbi algorithm.” *Proceedings of the IEEE*, 61(3), 268-278.

Fox, M.S., and Gruninger, M. (1998). “Enterprise modeling.” *AI Magazine*, 19(3):109.

Freund, Y., and Schapire, R.E. (1995). “Decision-theoretic generalization of on-line learning and an application to boosting.” *Computational Learning Theory*, 904:23-37.

Friedman, C., and Sideli, R. (1992). “Tolerating spelling errors during patient validation.” *Computers and Biomedical Research*, 25(5):486-509.

Fu Z., Christen P., and Zhou J. (2014) “A graph matching method for historical census household linkage.” *Advances in knowledge discovery and data mining*. 8443. Springer, Cham.

Ganganwar, V. (2012) “An overview of classification algorithms for imbalanced datasets.” *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42-47.

Ghodsi, A. (2006). “Dimensionality reduction a short tutorial.” Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada 37(2006):38.

Gimmer, A. E., (1984). *A glossary of historic masonry deterioration problems and preservation treatments*, Department of the Interior, Washington, D.C.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2006). “Ontological engineering: With examples from the areas of knowledge management.” e-*Commerce and the Semantic Web*, Springer Science & Business Media, Berlin, Germany.

Gong, R., and Chan, T.K.Y. (2006). “Syllable alignment: A novel model for phonetic string search.” *IEICE Transactions on Information and Systems*, 89(1):332-339.

Goyal, R., Whelan, M.J., and Cavalline, T.L. (2017). “Characterising the effect of external factors on deterioration rates of bridge components using multivariate proportional hazards regression.” *Structure and Infrastructure Engineering*, 13(7):894-905.

Graves, A., Mohamed, A.R., and Hinton, G. (2013). “Speech recognition with deep recurrent neural networks.” *Proc., 2013 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, IEEE, Piscataway, NJ, 6645-6649.

Gruber, T. R. (1995). “Toward Principles for the design of ontologies used for knowledge sharing.” *International Journal of Human-Computer Studies*, 43(5-6):907-928.

Guest, G., Bunce, A., and Johnson, L. (2006). “How many interviews are enough? An experiment with data saturation and variability.” Field Methods, 18(1):59-82.

Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., and Su, Z. (2009). “Domain adaptation with latent semantic association for named entity recognition.” *Proc., Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 281-289.

Guo, J., Che, W., and Yarowsky, D. (2015). “Cross-lingual dependency parsing based on distributed representations.” *Proc., 53rd Annual Meeting of the Association for*

Computational Linguistics and the 7th Intl. Joint Conf. on Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA,1234-1244.

Guo, J., Xu, G., Cheng, X., and Li, H. (2009). "Named entity recognition in query." *Proc., 32nd Int. ACM SIGIR Conf. Research and Developing in Information Retrieval*, Association for Computing Machinery, New York, NY, 267-274.

Gupta, R., and Sarawagi, S. (2009). "Answering table augmentation queries from unstructured lists on the web." *VLDB Endowment,* 2(1): 289-300.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). "Learning from class-imbalanced data: Review of methods and applications." *Expert Systems with Applications,* 73:220-239.

Halfawy, F.C., Hadipriono, F.C., Duane, J., and Larew, R. (2005). "Development of model based systems for integrated design of highway bridges." *Proc., Intl. Conf. on Civil, Structural and Environmental Engineering Computing*, Roma, Italy, 1-15.

Hall, J., Nilsson, J., and Nivre, J. (2010). "Single malt or blended? A study in multilingual parser optimization." *Trends in Parsing Technology*, 43:19-33.

Han, H., Wang, W.Y., and Mao, B.H. (2005). "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning." *Proc., Intl Conf. on Intelligent Computing*, Springer Science & Business Media, Berlin, Germany, 878-887.

Han, Y., Chen, W., Xiong, X., Li, Q., Qiu, Z., and Wang, T. (2019). "Wide & deep learning for improving named entity recognition via text-aware named entity normalization." *Proc., AAAI 2019 Workshop on Recommender Systems and Natural Language Processing*, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2015) "Semantic similarity from natural language and ontology analysis." *Synthesis Lectures on Human Language Technologies*, 8(1):1-254.

Hashimoto, K., Xiong, C., Tsuruoka, and Socher, R. (2017). "A joint many-task model: Growing a neural network for multiple NLP tasks." *Proc., 2017 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA.

Hassanzadeh, O. and Miller, R.J. (2009). "Creating probabilistic databases from duplicated data." *The International Journal on Very Large Data Bases*, 18(5):1141-1166.

Hassanzadeh, O., Chiang, F., Lee, H.C. and Miller, R.J. (2009). "Framework for evaluating clustering algorithms in duplicate detection." *VLDB Endowment*, 2(1):1282-1293.

Hatami, A., and Morcous, G. (2011). "Developing deterioration models for Nebraska bridges." *Final Rep. No. SPR-P1(11) M302*, Nebraska Department of Roads (NDOR), Lincoln, NE.

Haveliwala, T., Gionis, A., and Indyk, P. (2009). "Scalable techniques for clustering the Web." *Proc., 3rd Intl. Workshop on the Web and Databases*, Dallas, TX, 18-19.

He, H., and Garcia, E.A. (2008). "Learning from imbalanced data." *IEEE Transactions on Knowledge & Data Engineering*, (9):1263-1284.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." *Proc., 2008 IEEE Intl. Joint Conf. on Neural Networks*, IEEE, Piscataway, NJ, 1322-1328.

He, L., Song, Q., Shen, J., and Hai, Z. (2010). "Ensemble numeric prediction of nearest-neighbor learning." *Information Technology Journal*, 9(3):535-544.

He, Z., Jiang, W., and Chan, F.T. (2018). "Evidential supplier selection based on interval data fusion." *International Journal of Fuzzy Systems*, 20(4):1159-1171.

Henderson, J. (2004). "Discriminative training of a neural network statistical parser." *Proc., 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA.

Hobbs, D.W. (2011). "Concrete deterioration: Causes, diagnosis, and minimising risk." *International Materials Reviews*, 46(3):117-144.

Hinton, G. (2007). "Learning multiple layers of representation." *Trends in Cognitive Sciences,* 11(10):428-434.

Hobbs, J., and Riloff, E. (2010). "Information extraction." *Handbook of Natural Language Processing*, Chapman & Hall/CRC Press.

Hochreiter, S. (1998). "The vanishing gradient problem during learning recurrent neural nets and problem solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107-116.

Hochreiter, S., and Schmidhuber, J. (1997). "Long short-term memory." *Neural Computation*, 9(8):1735-1780.

Hofmann, T. (1999). "Probabilistic latent semantic indexing." *Proc., 22nd Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, 50-57.

Huang, L., and Sagae, K. (2010). "Dynamic programming for linear-time incremental parsing." *Proc., 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 1077-1086.

Huang, Y.-H. (2010). "Artificial neural network model of bridge deterioration." *Journal of Performance of Constructed Facilities*, 24(6):597-602.

Isozaki, H., and Kazawa, H. (2002). "Efficient support vector classifiers for named entity recognition." *Proc., 19th Intl. Conf. on Computational Linguistics,* Association for Computational Linguistics, Stroudsburg, PA, 1-7.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E. (1991) "Adaptive mixtures of local experts." *Neural Computation,* 3(1):79-87.

Jain, A. K., and Dubes, R.C. (1988). *Algorithms for clustering data*. Englewood Cliffs: Prentice Hall.

Jiang, J., and Zhai, C. (2007). "Instance weighting for domain adaptation in NLP." *Proc., 45th Annual Meeting of the Association of Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 26-271.

Jiang, J.J., and Conrath, D.W. (1997). "Semantic similarity based on corpus statistics and lexical taxonomy." arXiv preprint, cmp-lg/9709008.

Jiang, W., Xie, C., Zhuang, M., Shou, Y., and Tang, Y. (2016). "Sensor data fusion with z-numbers and its application in fault diagnosis." *Sensors*, 16(9):1509.

Jiang, Y., Lin, C., Meng, W., Yu, C., Cohen, A.M., and Smalheiser, N.R. (2014). "Rule-based deduplication of article records from bibliographic databases." *The Journal of Biological Database and Curation*, 2014(bat068): 1-7.

Jiao, F., Wang, S., Lee, C., and Greiner, R. (2006). "Semi-supervised conditional random fields for improved sequence segmentation and labeling." *Proc., 21st Intl. Conf. on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 209-216.

Jin, N. (2015). "NCSU-SAS-Ning: Candidate generation and feature engineering for supervised lexical normalization." *Proc., ACL 2015 Workshop on Noisy User-generated Text*, Association for Computational Linguistics, Stroudsburg, PA, 87-92.

Jones E., Oliphant E., and Peterson P. (2001). SciPy: Open Source Scientific Tools for Python. <http://www.scipy.org/> (Jun. 6, 2017).

Karapiperis, D., and Verykios, V.S. (2014). "A distributed near-optimal LSH-based framework for privacy-preserving record linkage." *Computer Science and Information Systems*, 11(2):745-763.

Karapiperis, D., and Verykios, V.S. (2015). "An LSH-based blocking approach with a homomorphic matching technique for privacy-preserving record linkage." *IEEE Transactions on Knowledge and Data Engineering*, 27(4):909-921.

Kejriwal M., and Miranker D.P. (2015). "Semi-supervised instance matching using boosted classifiers." *The semantic web*, Springer, Cham.

Keskustalo, H., Pirkola, A., Visala, K., Leppänen, E. and Järvelin, K. (2003). "Non-adjacent digrams improve matching of cross-lingual spelling variants." *Proc., 4th Intl. Semantic Web Conf.*, Manaus, Brazil, 252-265.

Khaleghi, B., Khamis, A., Karray, F.O., and Razavi, S.N. (2013). "Multisensor data fusion: A review of the state-of-the-art." *Information Fusion*, 14(1):28-44.

Kiperwasser, E., and Goldberg, Y. (2016). "Simple and accurate dependency parsing using bidirectional LSTM feature representations." *Transactions of the Association for Computational Linguistics*, (4):313-327.

Kubat, M., and Matwin, S. (1997). "Addressing the curse of imbalanced training sets: One-sided selection." *Proc., 14th Intl. Conf. on Machine Learning*, 179-186.

Kubota, S., and Mikami, I. (2013). "Development of product data model for maintenance in concrete highway bridges." *Applied Computational Intelligence and Soft Computing*, (2013)11:1-12.

Kudo, T., and Matsumoto, Y. (2002). "Japanese dependency analysis using cascaded chunking." *Proc., 6th Conf. on Natural Language L*earning, Taipei, Taiwan, 1-7.

Kuksa, P., and Qi, Y. (2010). "Semi-supervised bio-named entity recognition with word-codebook learning." *Proc., 2010 Society of Industrial and Applied Mathematics Intl. Conf. on Data Mining*, SIAM, Philadelphia, PA, 25-36.

Kuncoro, A., Ballesteros, M., Kong, L., Dyer, C., Neubig, G., and Smith, N.A. (2017). "What do recurrent neural network grammars learn about syntax?" *Proc., 55th Annual Conf. of the*

Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA.

Kurohashi, S., and Nagao, M. (1994). "KN parser: Japanese dependency/case structure analyzer." *Proc., Workshop on Sharable Natural Language Resources*, Nara, Japan 48-55.

LaDOTD. (2008). "Inspection of the Hale Boggs Memorial Bridge in Luling-Destrehan." Louisiana Department of Transportation and Development (LaDOTD), Baton Rouge, LA.

Lafferty, J., McCallum, A., and Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." *Proc., 18th Intl. Conf. on Machine Learning*, Williamstown, Massachusetts, 282-289.

Lahat, D., Adali, T., and Jutten, C. (2015). "Multimodal data fusion: An overview of methods, challenges, and prospects." *Proceedings of the IEEE*, 103(9):1449-1477.

Landauer, T.K., Foltz, P.W., and Laham, D. (1998). "An introduction to Latent Semantic Analysis." *Discourse Processes*, 25(2-3):259-284.

Laurikkala, J. (2001). "Improving identification of difficult small classes by balancing class distribution." *Proc., Conf. on Artificial Intelligence in Medicine in Europe*, Springer, Berlin, Germany, 63-66.

Le, T., and Jeong, H.D. (2017). "NLP-based approach to semantic classification of heterogeneous transportation asset data terminology." *Journal of Computing in Civil Engineering*, 31(6):04017057.

Leacock, C., and Chodorow, M. (1998). "Combining local context and WordNet similarity for word sense identification." *Wordnet: An Electronic Lexical Database*, 49(2):265-283.

Lee, H.C., Hsu, Y.Y., and Kao, H.Y. (2016). "AuDis: An automatic CRF-enhanced disease normalization in biomedical text." *Database*, 2016(bat091):1-11.

Lee, J., Guan, H., and Loo, Y. (2012). "Refinement of backward prediction method for reliable artificial intelligence-based bridge deterioration modelling." *Advances in Structural*, 15(5):825-836.

Lee, J.H., Guan, H., Loo, Y.C., Blumenstein, M., and Wang, X.P. (2011). "Modelling long-term bridge deterioration at structural member level using artificial intelligence techniques." *Applied Mechanics and Materials*, 444-453.

Leeman-Munk, S., Lester, J., and Cox, J. (2015). "NCSU_SAS_SAM: Deep encoding and reconstruction for normalization of noisy text." *Proc., ACL 2015 Workshop on Noisy User-generated Text,* Association for Computational Linguistics, Stroudsburg, PA, 154-161.

Lehnert, W., Cardie, C., Fisher, D., and Riloff, E. (1991). "University of Massachusetts: description of the CIRCUS system as ssed for MUC-3." *Proc., 3rd Conf. on Message Understanding,* San Diego, California, 223-233.

Lei, J., and Rinaldo, A. (2015) "Consistency of spectral clustering in stochastic block models." *The Annals of Statistics*, 43(1):215-237.

Levenshtein, V.I. (1966). "Binary codes capable of correcting deletions, insertions, and reversals." *Soviet Physics Doklady*, 10(8):707-710.

Li, D., Liu, C., and Gan, W. (2009). "A new cognitive model: Cloud model." *International Journal of Intelligent Systems*, 24(3):357-375.

Li, Q., Zhai, H., Deleger, L., and Lingren, T. (2013). "A sequence labeling approach to link medications and their attributes in clinical notes and clinical trial announcements for information extraction." *Journal of the American Medical Informatics Association*, 20(5):915-921.

Li, T., and Harris, D. (2019). "Automated construction of bridge condition inventory using natural language processing and historical inspection reports." *Proc., Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XIII,* International Society for Optics and Photonics, Bellingham, Washington, 109710T1-109710T8.

Li, Z., Ding, C., Wang, S., Wen, W., Zhuo, Y., Liu, C., Qiu, Q., Xu, W., Lin, X., Qian, X., and Wang, Y. (2019). "E-RNN: Design optimization for efficient recurrent neural networks in FPGAs." *Proc., 2019 IEEE Intl. Symposium on High Performance Computer Architecture*, IEEE, Piscataway, NJ, 69-80.

Liao, W., and Veeramachaneni, S. (2009). "A simple Semi-Supervised algorithm for named entity recognition." *Proc., NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, Boulder, Colorado, 58-65.

Lim, S., and Chi, S. (2019). "Xgboost application on bridge management systems for proactive damage estimation." *Advanced Engineering Informatics*, 41:100922.

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection." *Proc., IEEE Intl. Conf. on Computer Vision*, IEEE, Piscataway, NJ, 2980-2988.

Liu, D., and Nocedal, J. (1989). "On the Limited Memory BFGS method for large scale optimization." *Mathematical Programming*, 45(1-3):503-528.

Liu, F., Weng, F., and Jiang, X. (2012). "A broad-coverage normalization system for social media language." *Proc., 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, Association for Computational Linguistics, Stroudsburg, PA, 1035-1044.

Liu, J., Wang, C., Danilevsky, M., and Han, J. (2013). "Large-scale spectral clustering on graphs." *Proc., 23rd Intl. Joint Conf. on Artificial Intelligence*, Beijing, China, 1486-1492.

Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). "Recognizing named entities in Tweets." *Proc., 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Stroudsburg, PA, 359-367.

Liu, X., Zhou M., Wei F., Fu Z., and Zhou X. (2012). "Joint inference of named entity recognition and normalization for tweets." *Proc., 50th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 526-535.

Liu, Y., Liu, Y., and Chan, K. C. (2008). "Multiple video trajectories representation using double-layer isometric feature mapping." *Proc., 2008 IEEE Intl. Conf. on Multimedia and Expo*, IEEE, Piscataway, NJ, 129-132.

Long, B., Zhang, Z.M., Wu, X., and Yu, P.S. (2006) "Spectral clustering for multi-type relational data." *Proc., 23rd Intl. Conf. on Machine learning*, Pittsburgh, PA, 585-592.

Longadge, R., and Dongre, S. (2013). "Class imbalance problem in data mining review." arXiv preprint arXiv:1305.1707.

Lu, P., Pei, S., and Tolliver, D. (2016). "Regression model evaluation for highway bridge component deterioration using national bridge inventory data." *Journal of the Transportation Research Forum,* 51(1):5-16.

Lu, P., Wang, H., and Tolliver, D. (2019). "Prediction of bridge component ratings using ordinal logistic regression model." *Mathematical Problems in Engineering*, 2019(9797584):1-11.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). "An extensive experimental comparison of methods for multi-label learning." *Pattern Recognition,* 45(9):3084-3104.

Magdy, W., Darwish, K., Emam, O., and Hassan, H. (2007). "Arabic cross-document person name normalization." *Proc., 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Association for Computational Linguistics, Stroudsburg, PA, 25-32.

Mallapragada, P., Jin, R., and Jain, A. (2009). "Semiboost: Boosting for semi-supervised learning." *IEEE transactions on Pattern Analysis and Machine Intelligence,* 31(11):2000-2014.

Malouf, R. (2002). "A comparison of algorithms for maximum entropy parameter estimation." *Proc., 6th Conf. on Natural Language Learning*, Taipei, Taiwan, 1-7.

Mann, G., and McCallum, A. (2007). "Simple, Robust, Scalable Semi-Supervised Learning via Expectation Regularization." *Proc., 24th Intl. Conf. on Machine Learning,* Corvalis, Oregon, 593-600.

Mann, P.S. (1995). “Introductory statistics.” *Journal of the Royal Statistical Society-Series A Statistics in Society*, 158(2):339.

Manning, C., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

Marcus, M., Marcinkiewicz, M., and Santorini, B. (1993). “Building a large annotated corpus of english: The Penn Treebank.” *Computational Linguistics*, 18(2):313-330.

Marneffe, M., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. (2014). “Universal Stanford Dependencies: A cross-linguistic typology.*” Proc., 9th Intl. Conf. on Language Resources and Evaluation*, Reykjavik, Iceland, 4585-4592.

Mayberry. M., and Miikkulainen, R. (2005). “Broad-coverage parsing with neural networks.” *Neural Processing Letters*, 21(2):121-132.

McClain, J.O., and Rao, V.R. (1975). “Clustisz: A program to test for the quality of clustering of a set of objects.” *Journal of Marketing Research*, 12(4):456-460.

McDonald, R., and Nivre, J. (2007). “Characterizing the errors of data-driven dependency parsing models.” *Proc., 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech, 122-131.

Mehri, A., and Darooneh, A.H. (2011). “The role of entropy in word ranking.” *Physica A: Statistical Mechanics and its Applications*, 390(18-19):3157-3163.

Meila, M. (2016) “Spectral clustering: A Tutorial for the 2010’s.” Department of Statistics, University of Washington.

Mendenhall, W.M., and Sincich, T.L. (2016). *Statistics for engineering and the sciences*, Chapman and Hall/CRC, Boca Raton, Florida.

Mihalcea, R., and Csomai, A. (2007). “Wikify!: Linking documents to encyclopedic knowledge.” *Proc., 16th Conf. on Information and Knowledge Management*, Association for Computing Machinery, New York, NY, 233-242.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). "Corpus-based and knowledge-based measures of text semantic similarity." *Proc., 21st Intl. Conf. on Artificial Intelligence*, Boston, MA, 775-780.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J. (2013). "Distributed representations of words and phrases and their compositionality." *Proc., Neural Information Processing Systems Conference and Workshops*, Lake Tahoe, Nevada, 3111-3119.

Miller, S., Guinness, J., and Zamanian, A. (2004). "Name tagging with word clusters and discriminative training." *Proc., Human Language Technology Conf. of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 337-342.

Mishalani, R., and McCord, M. (2006). "Infrastructure condition assessment, deterioration modeling, and maintenance decision-making: New contributions for improved management." *Journal of Infrastructure Systems*, 12(3):145-146.

Mirza, M., Sommers, J., Barford, P., and Zhu, X. (2007). "A machine learning approach to TCP throughput prediction." *Proc., 2007 ACM SIGMETRICS Intl. Conf. on Measurement and Modeling of Computer Systems*, Association for Computing Machinery, New York, NY, 97-108.

MnDOT. (2006). "Fracture Critical Bridge Inspection In-Depth Report: I-35W over the Mississippi River at Minneapolis, Minnesota." Minnesota Department of Transportation (MnDOT), Saint Paul, MN.

Mohammadi, A., Javadi, S.H., Ciuonzo, D., Persico, V., and Pescapè, A. (2019). "Distributed detection with fuzzy censoring sensors in the presence of noise uncertainty." *Neurocomputing*, 351:196-204.

Mohar, B. (1997). "Some applications of Laplace eigenvalues of graphs." *Graph Symmetry*, 497:225-275.

Morcous, G. (2006). “Performance prediction of bridge deck systems using Markov chains.” *Journal of performance of Constructed Facilities*, 20(2):146-155.

Morcous, G., and Hatami, A. (2011). “Developing deterioration models for Nebraska bridges.” Final Reports & Technical Briefs from Mid-America Transportation Center.

Morcous, G., Rivard, H., and Hanna, A.M. (2002). “Modeling bridge deterioration using case-based reasoning.” *Journal of Infrastructure Systems*, 8(3):86-95.

Muller, P., Hathout, B., and Gaume, B. (2006). “Synonym extraction using a semantic distance on a dictionary.” *Proc. 1st Workshop on Graph Based Methods for Natural Language Processing*, Vancouver, BC, Canada, 65-72.

Muñoz, Y.F., Paz, A., De La Fuente-Mella, H., Fariña, J.V., and Sales, G.M. (2016). “Estimating bridge deterioration for small data sets using regression and markov models.” International *Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 10(5):663-670.

Nadeau, D., and Sekine, S. (2007). “A survey of named entity recognition and classification.” *Lingvisticae Investigationes*, 30(1):3-26.

Nadeau, D., Turney, P., and Matwin, S. (2006). “Unsupervised Named-entity recognition: Generating gazetteers and resolving ambiguity.” *Advances in Artificial Intelligence*, 4013(2006):266-277.

Nagelkerke, N.J. (1991). “A note on a general definition of the coefficient of determination.” *Biometrika*, 78(3):691-692.

NASEM. (2015). “Long-term bridge performance committee letter report: February 23, 2016.” National Academies of Sciences, Engineering, and Medicine (NASEM), Washington, D.C.: The National Academies Press.

NBIS. (2004). “National bridge inspection standards.” National Bridge Inspection Standards (NBIS), <http://www.fhwa.dot.gov/bridge/nbis.cfm >. (Dec. 2018).

NCHRP. (2011). “Framework for a National Database System for Maintenance Actions on Highway Bridges.” National Cooperative Highway Research Program (NCHRP), Washington, D.C.

NCHRP. (2014). “Proposed guideline for reliability-based bridge inspection practices.” National Cooperative Highway Research Program (NCHRP), Washington, D.C.

Ng, A.Y., Jordan, M.I., and Weiss, Y. (2002). “On spectral clustering: Analysis and an algorithm.” *Proc., 14th Intl. Conf. on Neural Information Processing Systems: Natural and Synthetic*, Vancouver, BC, Canada, 849-856.

Ng, V., and Cardie, C. (2002). “Improving machine learning approaches to coreference resolution.” *Proc., 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA,104-111.

Ngiam, J., Coates, A., and Lahiri, A. (2011). “On Optimization methods for deep learning.” *Proc., 28th Intl. Conf. on Machine Learning*, Bellevue, Washington, 265-272.

Nguyen, D.Q, Dras, M., and Johnson, M. (2017). “A novel neural network model for joint POS tagging and graph-based dependency parsing.” *Proc., 2017 SIGNLL Conf. on Computational Natural Language Learning*, Association for Computational Linguistics, Stroudsburg, PA.

Nivre, J. (2003). “An efficient algorithm for projective dependency parsing.” *Proc., 8th Intl. Workshop on Parsing Technologies*, Nancy, France, 149-160.

Nivre, J., and McDonald, R. (2008). “Integrating Graph-based and transition-based dependency parsers.” *Proc., 2008 Annual Conf. of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 950-958.

NOAA (2019). “Climate data online.” National Oceanic and Atmospheric Administration (NOAA), Silver Spring, MD, <https://www.ncdc.noaa.gov/cdo-web/> (Sep. 14, 2019).

Noy, N.F., and McGuinness, D.L. (2001). “Ontology development 101: A guide to creating your first ontology.” Stanford Knowledge Systems Laboratory.

NTSB. (2008). “Highway accident report Interstate 35W over the Mississippi River Minneapolis, Minnesota.” *No. NTSB/HAR-08/03*, National Transportation Safety Board (NTSB) Washington, D.C.

Oflazer, K. (2003). “Dependency parsing with an extended finite-state approach.” *Computational Linguistics*, 29(4):515-544.

Olson, D.L., and Delen, D. (2008). *Advanced data mining techniques*, Springer Publishing Company, New York, NY.

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

Paassen, B., Stöckel, A., Dickfelder, R., and Göpfert, J. (2014). “Ontology-based extraction of structured information from publications on preclinical experiments for spinal cord injury treatments.” *Proc., 3rd Workshop on Semantic Web and Information Extraction*, Dublin, Ireland, 25-32.

Pafilis, E., Frankild, S.P., Fanini, L., Faulwetter, S., Pavloudi, C., Vasileiadou, A., Arvanitidis, C., and Jensen, L.J. (2013). “The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text.” *PLoS One*, 8(6):e65390.

Pathria, R.K., and Beale, P. (2011). *Statistic mechanics*, Academic Press, Cambridge, MA.

Pawlak, Z., and Skowron, A. (1992). “Rudiments of rough sets.” *Information Sciences*, 177(1):3-27.

PCA. (2002). “Types and causes of concrete deterioration.” Portland Cement Association (PCA), <http://www.cement.org/docs/default-source/th-paving-pdfs/concrete/types-and-causes-of-concrete-deterioration-is536.pdf> (Sep. 20, 2015).

PCA. (2016). “Effects of Substances on Concrete and Guide to Protective Treatment.” Portland Cement Association (PCA), Skokie, Illinois.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., and Vanderplas, J. (2011). "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research*, 12:2825-2830.

Pei, W., Ge, T., and Chang, B. (2015). "An effective neural network model for graph-based dependency parsing." *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Intl. Joint Conference on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 313-322.

Pes, B., Dessì, N., and Angioni, M. (2017). "Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data." *Information Fusion*, 35:132-147.

Phua, C., Smith-Miles, K., Lee, V., and Gayler, R. (2012). "Resilient identity crime detection." *IEEE Transactions on Knowledge and Data Engineering*, 24(3):533-546.

Priya, R., and Aruna, P. (2012). "SVM and neural network based diagnosis of diabetic retinopathy." *International Journal of Computer Applications*, 41(1):6-12.

Protégé. (2016). "Protégé 3.4.5." <http://protege.stanford.edu/download/protege/old-releases/Protege 3.x/3.4.5/installanywhere/Web_Installers/>.

Puhlmann S., Weis M., and Naumann F. (2006). "XML duplicate detection using sorted neighborhoods." *Advances in database technology*, Springer, Berlin, Germany.

Python Core Team (2015). Python: A Dynamic, Open Source Programming Language. <http://www.python.org/> (Jun. 6, 2017).

Qi, Y., Das, S., Collobert, R., and Weston, J. (2014). "Deep learning for character-based information extraction." *Proc., 36th European Conf. on Information Retrieval Research*, Amsterdam, Netherlands, 668-674.

Qiao, Y., Moomen, M., Zhang, Z., Agbelie, B., Labi, S., and Sinha, K.C (2016). "Modeling deterioration of bridge components with binary probit techniques with random effects." *Transportation Research Record*, 2550(1):96-105.

Ranjith, S., and Setunge, S. (2011). “Deterioration prediction of timber bridge elements using the Markov chain.” *Journal of Performance of Constructed Facilities*, 27(3):319-325.

Resnik, P. (1995). “Using information content to evaluate semantic similarity in a taxonomy.” *Proc., 14th Intl. Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada.

Riloff, E. (1993). “Automatically constructing a dictionary for information extraction tasks.” *Proc., 11th National Conf. on Artificial Intelligence*, Washington, D.C., 811-816.

Ritter, A., Etzioni, O., and Clark, S. (2012). “Open domain event extraction from Twitter.” *Proc., 18th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Association for Computational Linguistics, Stroudsburg, PA, 1104-1112.

Rogers, H. (2006). “Bridge preservation in PennDOT.” Pennsylvania Department of Transportation (DOT), Washington, D.C.

Roweis, S.T., and Saul, L.K. (2000). “Nonlinear dimensionality reduction by locally linear embedding.” *Science*, 290(5500):2323-2326.

Rumelhard, D.E., Hinton, G.E., and Williams, R.J. (1986). “Learning representations by back-propagating errors.” *Nature*, 323:533-536.

Sagae, K., and Lavie, A. (2006). “Parser combination by reparsing.” *Proc., Human Language Technology Conference of the NAACL*, New York, NY, 129-132.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2017). “Recent advances in recurrent neural networks.” arXiv preprint arXiv:1801.01078.

Samko, O., Marshall, A.D., and Rosin, P.L. (2006). “Selection of the optimal parameter value for the Isomap algorithm.” *Pattern Recognition Letters*, 27(9):968-979.

Samuelsson, C. (2000). “A statistical theory of dependency syntax.” *Proc., 18th Conf. on Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 684-690.

Sarawagi, S. (2008). “Information extraction.” *Foundations and Trends in Databases*, 1(3):261-377.

Saul, L.K., Weinberger, K.Q., Ham, J.H., Sha, F., and Lee, D.D. (2006). “Spectral methods for dimensionality reduction.” *Semisupervised Learning*, 293-308.

Schapire, R.E. (1990). “The strength of weak learnability.” *Machine Learning*, 5(2):197-227.

Schiele, B. (2002). “How many classifiers do I need?” *Proc., 16th Intl. Conf. on Pattern Recognition*, Institute of Electrical and Electronics Engineers, New York, NY.

Sekine, S., Grishman, R., and Shinnou, H. (1998). “A decision tree method for finding and classifying names in Japanese texts.” *Proc., 6th Workshop on Very Large Corpora*, Montreal, Quebec, Canada, 171-178

Sentz, K., and Ferson, S. (2002). “Combination of evidence in Dempster-Shafer theory.” Sandia Report, *SAND2002-0835*, Sandia National Laboratories, Albuquerque, NM.

Shafer, G. (1976). “A mathematical theory of evidence.” Princeton University Press, Princeton, NJ.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2011). “Pegasos: Primal estimated sub-gradient solver for SVM.” *Mathematical*, 127(1):3-30.

Shannon, C.E. (1948). “A mathematical theory of communication.” *Bell System Technical Journal*, 27(3):379-423.

Shi, J., and Malik, J. (2000). “Normalized cuts and image segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888-905.

Shibuya, N., Nukala, B. T., Rodrigues, A.I., Tsay, J., Nguyen, T.Q., Zupancic, S., and Lie, D.Y.C. (2015). “A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (SVM) classifier.” *Proc., 8th Intl. Conf. on Mobile Computing and Ubiquitous Networking*, Hokkaido, Japan, 66-67.

Shinyama, Y., and Sekine, S. (2004). “Named Entity discovery using comparable news articles.” *Proc., 20th Intl. Conf. on Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA.

Silva, V.D., and Tenenbaum, J.B. (2003). “Global versus local methods in nonlinear dimensionality reduction.” *Proc., Advances in Neural Information Processing Systems*, 721-728.

Singla, P., and Domingos, P. (2006). “Entity resolution with Markov logic.” *Proc., 6th Intl. Conf. on Data Mining*, Hong Kong, China, 572-582.

Smith, T.F., and Waterman, M.S. (1981). “Identification of common molecular subsequences.” *Journal of Molecular Biology*, 147(1):195-197.

Smola, A., and Vapnik, V. (1997). “Support vector regression machines.” *Advances in Neural Information Processing Systems*, 9:155-161.

Song, C., Zhao, H., Xu, Z., and Hao, Z. (2019). “Interval-valued probabilistic hesitant fuzzy set and its application in the Arctic geopolitical risk evaluation.” *International Journal of Intelligent Systems*, 34(4):627-651.

Soon, W.M, Ng, H.T., and Lim, D.C.Y. (2006) “A machine learning approach to coreference resolution of noun phrases.” *Computational Linguistics*, 27(4):521-544.

Soysal, E., Cicekli, I., and Baykal, N. (2010). “Design and evaluation of an ontology based information extraction system for radiological reports.” *Computers in Biology and Medicine*, 40(11-12):900-911.

Stoilos, G., Stamou, G., and Kollias, S. (2005). “A string metric for ontology alignment.” *Proc., 4th Intl. Semantic Web Conf.*, Galway, Ireland, 624-637.

Strubell, E., and McCallum A. (2017). “Dependency parsing with dilated iterated graph CNNs.” *Proc., 2nd Workshop on Structured Prediction for Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA.

Sun, S. (2013). “A survey of multi-view machine learning.” *Neural Computing and Applications*, 23(7):2031-2038.

Sun, Y., and Han, J. (2012). “Mining heterogeneous information networks: Principles and methodologies.” *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1-159.

Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. (2011). “PathSim: meta path-based top-k similarity search in heterogeneous information networks.” *Proc., 37th Intl. Conf. on Very Large Data Bases*, Seattle, Washington, 992-1003.

Sun, Y., Kamel, M.S., and Wang, Y. (2006). “Boosting for learning multiple classes with imbalanced class distribution.” *Proc., 6th Intl. Conf. on Data Mining*, Institute of Electrical and Electronics Engineers, New York, NY, 592-602.

Sutskever, I., Martens, J., and Hinton, G.E. (2011). “Generating text with recurrent neural networks.” *Proc., 28th Intl. Conf. on Machine Learning*, Bellevue, WA, 1017-1024.

Sutton, C., and McCallum, A. (2006). “An Introduction to Conditional Random Fields for Relational Learning.” *Introduction to Statistical Relational Learning*, MIT Press, Cambridge, MA.

Swartout, B., Patil, R., Knight, K., and Russ, T. (1996). “Toward distributed use of large-scale ontologies.” *Proc., 10th Workshop on Knowledge Acquisition for Knowledge-Based Systems*, AAAI, Menlo Park, CA, 138-148.

Tahmasebi, A.M., Zhu, H., Mankovich, G., Prinsen, P., Klassen, P., Pilato, S., van Ommering, R., Patel, P., Gunn, M.L., and Chang, P. (2019). “Automatic normalization of anatomical phrases in radiology reports using unsupervised learning.” *Journal of Digital Imaging*, 32(1):6-18.

Tang, B., Cao, H., Wu, Y., Jiang, M., and Xu, H. (2012). “Clinical entity recognition using structural support vector machines with rich features.” *Proc., ACM 6th Intl. Workshop on Data and Text Mining in Biomedical Informatics*, Association for Computational Linguistics, Stroudsburg, PA, 13-20.

Tang, J., Belletti, F., Jain, S., Chen, M., Beutel, A., Xu, C., and H Chi, E. (2019). "Towards neural mixture recommender for long range dependent user sequences." *Proc., World Wide Web Conf.,* Association for Computing Machinery, New York, NY, 1782-1793.

Tapanainen, P., and Järvinen, T. (1997). "A non-projective dependency parser." *Proc., 5th Conf. on Applied Natural Language Processing*, Washington, D.C., 61-71.

Tenenbaum, J.B., De Silva, V., and Langford, J.C. (2000). "A global geometric framework for nonlinear dimensionality reduction." *Science,* 290(5500):2319-2323.

Tian, W., de Wilde, P., Li, Z., Song, J., and Yin, B. (2018). "Uncertainty and sensitivity analysis of energy assessment for office buildings based on Dempster-Shafer theory." *Energy Conversion and Management,* 174:705-718.

Tongco, M. D. C. (2007). "Purposive sampling as a tool for informant selection." *Ethnobotany Research and Applications,* 2007(5):147-158.

Torra, V. (2010). "Hesitant fuzzy sets." *International Journal of Intelligent Systems*, 25(6):529-539.

Toutanova, K., Klein, D., Manning, C.D., and Singer, Y. (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network." *Proc., 2003 Conf. of the North American chapter of the Association for Computational Linguistics on Human Language Technology*, Association for Computational Linguistics, Stroudsburg, PA, 173-180.

TRB. (2015). "Transportation research thesaurus." Transportation Research Board (TRB), <http://trt.trb.org/trt.asp?> (Oct. 25, 2015).

Trinh, T.H., Dai, A.M., Luong, M.T., and Le, Q.V. (2018). "Learning longer-term dependencies in RNNs with auxiliary losses." arXiv preprint arXiv:1803.00144.

Turney, P.D. (2001). "Mining the Web for synonyms: PMI-IR versus LSA on TOEFL." *Proc., 12th European Conf. on Machine Learning*, Freiburg, Germany, 491-502.

Ulieru, M., and Madani, S.A. (2006). "An application of industrial agents to concrete bridge monitoring." *Proc., 3rd Intl. Conf. on Informatics in Control, Automation and Robotics*, IEEE, Piscataway, NJ.

USACE. (1995). "evaluation and repair of concrete structures." *Report No. 20314-1000*, Department of the Army U.S. Army Corps of Engineers (USACE), Washington, D.C.

Varga, O., Nagy, I.G., Burai, P., Tomor, T., Lénárt, C., and Szabó, S. (2018). "Land cover analysis based on descriptive statistics of Sentinel-2 time series data." *Acta Geographica Debrecina Landscape & Environment*, 12(2):1-9.

Vatsalan, D., Christen, P., and Verykios, V.S. (2013). "A taxonomy of privacy-preserving record linkage techniques." *Information Systems*, 38(6):946-969.

Von Luxburg, U. (2007). "A tutorial on spectral clustering." *Statistics and Computing*, 17(4):395-416.

Vrandečić, D. (2009). *Ontology evaluation*. Springer, Berlin, Germany.

Wagner, J., and Foster, J. (2015). "DCU-ADAPT: Learning edit operations for microblog normalisation with the generalised perceptron." *Proc., ACL 2015 Workshop on Noisy User-generated Text*, Association for Computational Linguistics, Stroudsburg, PA, 93-98.

Wang, F., Wu, W., Li, Z., and Zhou, M. (2017). "Named entity disambiguation for questions in community question answering." *Knowledge-Based Systems*, 126:68-77.

Wang, W., and Harper, M.P. (2004). "A statistical constraint dependency grammar (CDG) parser." *Proc., Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, Barcelona, Spain, 42-49.

Washer, G., Connor, R., Ciolko, A., Kogler, R., Fish, P., and Forsyth, D. (2014). "Proposed guideline for reliability-based bridge inspection practices." *NCHRP Report 782*, National Cooperative Highway Research Program (NCHRP), Washington, D.C., 1-220.

Wei, Q., Chen, T., Xu, R., He, Y., and Gui, L. (2016). "Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks." *Database*, 2016(baw140).

Weis, M., and Naumann, F. (2004). "Detecting duplicate objects in XML documents." *Proc., 2004 Intl. Workshop on Information Quality in Information Systems*, Paris, France, 10-19.

Weiss, D., Alberti, C., Collins, M., and Petrov, S. (2015). "Structured training for neural network transition-based parsing." *Proc., 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Intl. Joint Conf. on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 323-333.

Wellalage, N.K.W., Zhang, T., and Dwight, R. (2014). "Calibrating Markov chain-based deterioration models for predicting future conditions of railway bridge elements." *Journal of Bridge Engineering*, 20(2):04014060.

Wimalasuriya, D., and Dou, D. (2010). "Ontology-based information extraction: An introduction and a survey of current approaches." *Journal of Information Science*, 36(3):306-323.

Wimmer, M., Schuller, B., Arsic, D., Radig, B., and Rigoll, G. (2008). "Low-level fusion of audio and video feature for multi-modal emotion recognition." *Proc., 3rd Intl. Conf. on Computer Vision Theory and Applications*, Science and Technology Publications, New York, NY, 145-151.

Winkler, W.E., and Thibaudeau, Y. (1991). "An application of the Fellegi-Sunter model of record linkage to the 1990 US Decennial Census." US Bureau of the Census, Washington D.C.

Winkler, W.E. (2006). "Overview of record linkage and current research directions." Statistical Research Division, Bureau of the Census, Washington D.C.

Wolpert, D.H. (1992). "Stacked generalization." *Neural Networks*, 5(2):241259.

Wong, W., Liu, W., and Bennamoun, M. (2012). "Ontology learning from text: A look back and into the future." *ACM Computing Surveys (CSUR)*, 44(4):20.

Woodson, R. D. (2009). *Concrete structures: Protection, repair and rehabilitation*, Butterworth-Heinemann, Oxford, UK.

WSDOT. (2009). "Bridge inspection reports: South Park Bridge." Washington Department of Transportation (WSDONT), Olympia, WA.

WSDOT. (2013). "2013 Bridge inspection report for bridge 0005115A." Washington Department of Transportation (WSDONT), Olympia, WA.

WSDOT. (2015). "2015 Bridge inspection report for bridge 0005115A." Washington Department of Transportation (WDOT), Olympia, WA.

WSDOT. (2016). "2016 Bridge inspection report for bridge 0000965A." Washington Department of Transportation (WSDOT), Olympia, WA.

WSDOT. (2019) "Traffic data GeoPortal." Washington Department of Transportation (WSDOT), Olympia, WA, <https://www.wsdot.wa.gov/mapsdata/tools/trafficplanningtrends.htm> (Sep. 14, 2019).

WiSDOT. (2015). "Maintenance and operations." Wisconsin Department of Transportation (WisDOT), Madison, WI.

Wu, D., Lee, W., Ye, N., and Chieu, H. (2009). "Domain adaptive bootstrapping for named entity recognition." *Proc., 2009 Conf. on Empirical Methods in Natural Language Processing*, Singapore, 1523-1532.

Wu, F., and Weld, D. (2010). "Open information extraction using Wikipedia." *Proc., 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 118-127.

Wu, S.M., You, X.Y., Liu, H.C., and Wang, L.E. (2018). "Improving quality function deployment analysis with the cloud MULTIMOORA method." *International Transactions in Operational Research*, 0(2017):1-11

Wu, Z., and Palmer, M. (1994). "Verbs semantics and lexical selection." *Proc., 32nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 133-138.

Xiao, F. (2019). "Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy." *Information Fusion*, 46:23-32.

Xie, S., and Liu, Y. (2008). "Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization." *Proc., 2008 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, IEEE, Piscataway, NJ, 4985-4988.

Xu, C., Tao, D., and Xu, C. (2013). "A survey on multi-view learning." *Computing Research Repository*, arXiv:1304.5634.

Xu, H., Stenner, S., and Doan, S. (2010). "MedEx: A medication information extraction system for clinical narratives." *Journal of the American Medical Informatics Association*, 17(1):19-24.

Yamada, H., and Matsumoto, Y. (2003). "Statistical dependency analysis with support vector machines." *Proc.,8th Intl. Conf. on Parsing Technologies*, Association for Computational Linguistics, Stroudsburg, PA, 195-206.

Yang, Y., and Eisenstein, J. (2013). "A log-linear model for unsupervised text normalization." *Proc., 2013 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 61-72.

Yang, Y., and Webb, G.I. (2001). "Proportional k-interval discretization for naive-Bayes classifiers." *Proc., European Conf. on Machine learning*, Springer, Berlin, Germany, 564-575.

Yazdani, M., and Henderson, J. (2015). "Incremental recurrent neural network dependency parser with search-based discriminative training." *Proc., 19th Conf. on Computational Language Learning*, Beijing, China, 142-152.

Yu, Z., Zhang, Y., You, J., Chen, C.P., Wong, H.S., Han, G., and Zhang, J. (2017). "Adaptive semi-supervised classifier ensemble for high dimensional data classification." *IEEE Transactions on Cybernetics*, 99:1-14.

Zadeh, L.A. (1965). "Fuzzy sets." *Information and Control*, 8(3):338-353.

Zadeh, L.A. (1978). "Fuzzy sets as a basis for a theory of possibility." *Fuzzy Sets and Systems*, 1(1):3-28.

Zadeh, M.R., Amin, S., Khalili, D., and Singh, V.P. (2010). "Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions." *Water Resources Management*, 24(11):2673-2688.

Zambon, I., Vidovic, A., Strauss, A., Matos, J., and Amado, J. (2017). "Comparison of stochastic prediction models based on visual inspections of bridge decks." *Journal of Civil Engineering and Management*, 23(5):553-561.

Zhang, C. (2015). "Applying data fusion techniques for benthic habitat mapping and monitoring in a coral reef ecosystem." *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:213-223.

Zhang, C., and Ma, Y. (2012). "Ensemble learning." *Ensemble Machine Learning: Methods and Applications*, Springer, New York, NY.

Zhang, J., and El-Gohary, N. (2013). "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking." *Journal of Computing in Civil Engineering*, 30(2):0000346

Zhang, J., and El-Gohary, N.M. (2016). "Extending building information models semiautomatically using semantic natural language processing techniques." *Journal of Computing in Civil Engineering*, 30(5):C4016004.

Zhang, L., Wu, X., Zhu, H., and AbouRizk, S.M. (2017). "Perceiving safety risk of buildings adjacent to tunneling excavation: An information fusion approach." *Automation in Construction*, 73:88-101.

Zhang, X., Jiao, L., Liu, F., Bo, L., and Gong, M. (2008). "Spectral clustering ensemble applied to SAR image segmentation." *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2126-2136.

Zhang, Y., and Clark, S. (2008). "A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing using beam-search." *Proc., Conf. on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, 562-571.

Zhang, Y., and Nivre, J. (2011). "Transition-based dependency parsing with rich non-local features." *Proc., 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Stroudsburg, PA, 188-193.

Zheng, H., and Deng, Y. (2018). "Evaluation method based on fuzzy relations between Dempster-Shafer belief structure." *International Journal of Intelligent Systems*, 33(7):1343-1363.

Zhou W., Yu C., Smalheiser N., Torvik V., and Hong J. (2007). "Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature." *Proc., 30th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, 655-662.

Zhou, H., Zhang, Y., and Huang, S. (2015). "A neural probabilistic structured-prediction model for transition-based dependency parsing." *Proc., 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Intl. Joint Conf. on Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, 1213-1222.

Zhou, P., and El-Gohary, N. (2015). "Ontology-Based multilabel text classification of construction regulatory documents." *Journal of Computing in Civil Engineering*, 30(4):0000530

Zhou, P., and El-Gohary, N. (2015). "Ontology-based information extraction from environmental regulations for supporting environmental compliance checking." *Proc. 2015 Intl. Workshop on Computing in Civil Engineering*, ASCE, Reston, VA, 190-198.

Zhou, P., and El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." *Automation in Construction* 74:103-117.

Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R., and Rosen, M.S. (2018). “Image reconstruction by domain-transform manifold learning.” *Nature,* 555(7697):487.

APPENDIX A: LIST OF DATA FEATURES

The features of the National Bridge Inventory (NBI) data, the National Bridge Elements (NBE) data, the traffic data, the weather data, and the data extracted from the textual bridge inspection reports are presented in Table A.1.

Table A.1. Data features.

Data	Features
National Bridge Inventory (NBI) data	State code, structure number, inventory route, record type, route signing prefix, designated level of service, route number, directional suffix, highway agency district, county (parish) code, place code, features intersected, features intersected, critical facility indicator, facility carried by structure, location, inventory route minimum vertical clearance, kilometer point, base highway network inventory route sub route number, linear referencing system inventory route, sub route number, latitude, longitude, bypass/detour length, toll, maintenance responsibility, owner, functional class of inventory route, year built, lanes on/under structure, lanes on structure, lanes under structure, average daily traffic, year of average daily traffic, design load, approach roadway width, bridge median, skew, structure flared, traffic safety features, bridge railings, transitions, approach guardrail, approach guardrail ends, historical significance, navigation control, navigation vertical clearance, navigation horizontal clearance, structure open/posted/closed, type of service, type of service on bridge, type of service under bridge, structure type, main, kind of material/design, type of design/construction, structure type, approach spans, kind of material/design, type of design/construction, number of spans in main unit, number of approach spans, inventory route total horizontal clearance, length of maximum span, structure length, curb/sidewalk widths, left curb/sidewalk width, right curb/sidewalk width, bridge roadway width curb-to-curb, deck width out-to-out, minimum vertical clearance over bridge roadway, minimum vertical under clearance, reference feature, minimum vertical under clearance, minimum lateral under clearance on right, reference feature, minimum lateral under clearance, minimum lateral under clearance on left, deck, superstructure, substructure, channel/channel protection, culverts, method used to determine operating rating, operating rating, method used to determine inventory rating, inventory rating, structural evaluation, deck geometry, under clearance vertical & horizontal, bridge posting, waterway adequacy, approach roadway alignment, type of work, type of work proposed, work done by, length of structure improvement, inspection date, designated inspection frequency, critical feature inspection, fracture critical details, underwater inspection, other special inspection, critical feature inspection dates, fracture critical details date, underwater inspection date, other special inspection date, bridge improvement cost, roadway improvement cost, total project cost, year of improvement cost estimate, border bridge, neighboring state code, percent responsibility, border bridge structure number, STRAHNET highway designation, parallel structure designation, direction of traffic, temporary structure designation, highway system of inventory route, federal lands highways, year reconstructed, deck structure type, wearing surface/protective system, type of wearing surface, type of membrane, deck protection, average daily truck traffic, designated national network, pier/abutment protection, NBIS bridge length, scour critical bridges, future average daily traffic, year of future average daily traffic, and minimum navigation vertical clearance vertical lift bridge.

Table A.1. Data features (cont'd).

Data	Features
National Bridge Elements (NBE) data [1]	Concrete deck, bridge deck surface, fully supported concrete deck, post tensioned concrete deck, concrete deck - lightweight aggregate, concrete deck w/coated bars, steel orthotropic deck, steel deck - concrete filled grid, deck - corrugated or other steel system, timber deck, fiber reinforced polymer (FRP) - deck, concrete deck soffit, deck rebar cover flag, concrete slab, concrete hollow slab, prestressed concrete slab, prestressed concrete slab w/coated bars, concrete slab w/coated bars, timber slab, prestressed concrete girder w/coated strands, steel rolled girder, steel riveted girder, steel welded girder, concrete encased steel girder, prestressed concrete trapezoidal girder, thin flange girder, post tensioned concrete segmental box girder, steel box girder, prestressed concrete super girder, post tension concrete box girder, concrete box girder, steel open girder, prestressed concrete bulb-t girder, prestressed concrete multiple web girder units, concrete girder, timber glue-lam girder, steel stringer, concrete multiple web girder unit, prestressed concrete girder, concrete stringer, timber sawn girder, timber stringer, concrete truss, steel thru truss, steel deck truss, truss gusset plates, timber truss, timber arch, steel arch, steel tied arch, steel suspender, concrete arch, earth filled concrete arch, suspension - main cable, suspension - suspender cable, cable stayed bridge - cable, concrete column on spandrel arch, steel floor beam, prestressed concrete floor beam, concrete floor beam, timber floor beam, steel column on spandrel arch, steel hanger, steel pin, tension hold down anchor assembly, abutment fill, steel pile/column, prestressed hollow concrete pile/column, prestressed concrete pile/column, concrete pile/column, timber pile/column, concrete pile/column w/steel jacket, concrete pile/column w/composite wrap, submerged concrete pile/column w/steel jacket, concrete pier wall, other pier wall, concrete submerged pier wall, other submerged pier wall, concrete web wall between columns, concrete abutment, timber abutment, other abutment, steel abutment, concrete cantilevered span abutment, concrete submerged foundation, concrete foundation, timber foundation, steel submerged pile/column, prestressed concrete submerged pile/column, concrete submerged pile/column, timber submerged pile/column, timber cap rehab with steel, steel pier cap/crossbeam, submerged hollow prestressed concrete pile/column, prestressed concrete pier cap/crossbeam, concrete pier cap/crossbeam, timber pier cap, concrete floating pontoon, pontoon hatch/bulkhead, floating bridge - anchor cable, metal culvert, concrete culvert, timber culvert, other culvert, steel open grid sidewalk and supports, steel concrete filled grid sidewalk and supports, corrugated/orthotropic sidewalk and supports, concrete sidewalk and supports, fiber reinforced polymer (FRP) sidewalk and supports, elastomeric bearing, moveable bearing, concealed bearing or bearing system, fixed bearing, pot bearing, disc bearing, isolation bearing, concrete roadway approach slab, bridge impact, metal bridge railing, concrete bridge railing, timber bridge railing, other bridge railing, timber pedestrian rail, other pedestrian rail, metal pedestrian rail, concrete pedestrian rail, damaged bolts or rivets, steel cracking, pack rust, bridge movement, scour, movable bridge, seismic pier crossbeam bolster, seismic pier infill wall, seismic - longitudinal restrainer, seismic - transverse restrainer, seismic - link/pin restrainer, seismic - catcher block, seismic - column silo, cathodic protection, concrete deck delamination testing, primary safety inspection, secondary safety inspection, asphalt butt joint seal, asphalt open joint seal, strip seal - welded, bolt down - sliding plate w/springs, bolt down panel - molded rubber, assembly joint seal (modular), silicone rubber joint filler, asphalt plug, steel angle w/raised bars, joint paved over flag, concrete slab in-span joint, flexible joint seal, open concrete joint, concrete bulb-t, compression seal/concrete header, compression seal/polymer header, compression seal/steel header, steel angle header, steel sliding plate, steel sliding plate w/raised bars, steel fingers, steel fingers w/raised bars, strip seal - anchored, movable bridge steel tower, ceramic tile, bridge mounted sign structures, bridge luminaire pole and base, fender system/pier protection, polyester concrete overlay, AC over a polymer overlay, BST on concrete (chip seal), asphalt concrete (AC) overlay w/high performance membrane, red lead alkyd paint system, inorganic zinc/vinyl paint system, inorganic zinc/urethane paint system, organic zinc/urethane paint system, coal tar epoxy paint system, metalizing, galvanizing, epoxy paint for weathering steel, zinc primer, weathering steel patina.

Table A.1. Data features (cont'd).

Traffic data	Average daily traffic, percentage of single unit trucks, percentage of double unit trucks, and percentage of triple unit trucks.
Weather data	Cooling degree day normal (with 45, 50, 55, 57, 60, 65, 70, and 72 ^{o}F bases), heating degree day normal (40, 45, 50, 55, 57, 60, and 65 ^{o}F base), annual precipitation totals, number of days during which they year with precipitation is greater than 0.01, 0.10, 0.50, and 1.00 inches, annual snowfall totals, number of days during the year snowfall is greater than 0.1, 1.0, 3.0, 5.0, and 10.0 inches, number of days during the year snow depth is greater than 1.0, 3.0, 5.0, and 10.0 inches, diurnal temperature range, annual average temperature, annual maximum temperature, number of days per year where the maximum temperature is greater than or equal to 40, 50, 60, 70, 80, 90, and 100 ^{o}F, annual minimum temperature, and number of days per year where the minimum temperature is less than or equal to 0, 10, 20, 32, 40, 50, 60, and 70 ^{o}F.
Textual bridge inspection reports	Bridge element, deficiency, deficiency cause, maintenance action, maintenance material, numerical measure, numerical measure unit, categorical quantity measure, categorical severity measure, and date.

[1] Each element is associated with four quantities: total quantity, quantity in condition state "good", quantity in condition state "fair", quantity in condition state "poor", and quantity in condition state "severe".