

Mining Domain Knowledge from Unstructured Multi-modal Data for Smart Bridge Infrastructure Management

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

Tianshu Li

May 2021

ACKNOWLEDGEMENTS

Throughout the work of this dissertation, I have received a great deal of support and assistance.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Devin Harris, for his guidance in my research, and the motivation and encouragement in pursuing my research interest. His expertise was invaluable in formulating the research questions, and I could not have completed a fulfilling Ph.D. study without his support.

I also want to express my gratitude to my Ph.D. committee, including Professor Arsalan Heydari, and Professor Michael Porter from the Department of Engineering Systems and Environment, Professor Vicente Ordóñez from the Department of Computer Science at the University of Virginia, and Professor David Lattanzi from George Mason University for their valuable feedback that contributed to the quality of this dissertation.

I would like to acknowledge my colleagues and teammates at the Mobile Laboratory for Rapid Evaluation of Transportation Infrastructure for the positive atmosphere and the inspiring work. I would particularly like to single out my former colleague and great friend Dr. Mohamad Alipour for his guidance and collaboration along the path of forming this dissertation.

I also want to acknowledge the help of Dr. Bernie Kassner, Dr. Bridget Donaldson from Virginia Transportation Research Council (VTRC), and Mr. John Lindemann from the Virginia Department of Transportation (VDOT) in accessing and collecting the inspection report data.

Besides, I would like to thank all my friends at the University of Virginia, especially Yichen Jiang and Zhennan Zhu, who provided warm company as well as happy distractions to rest my mind outside research. I would also like to thank my parents for their wise counsel and tremendous understanding. Last but not least, thanks to my husband and best friend, Dr. Wei Ma, for supporting and encouraging me to leap for a better self.

This dissertation is finished during the pandemic of COVID-19, at the end of the second remote semester. The past year has been a difficult one, to be physically apart from my friends and family, but I am grateful that we all stayed healthy, were able to continue our works, and are towards the end of the isolation caused by the pandemic.

ABSTRACT

The management of the bridge system encloses a pipeline of condition data collection (*i.e.* inspections), condition assessment, and deterioration prediction. Visual inspections are performed periodically, submitting ratings of bridge conditions to the National Bridge Inventory (NBI) and the element-level rating data to support deterioration prediction. While the current experience-driven inspection and assessment has been practiced for decades, today's bridge infrastructure system in the U.S. faces critical preservation challenges in its sheer volume of aging and deteriorating bridges with limited funding and resources. Improving the efficiency of the bridge infrastructure management in the face of critical preservation challenges calls for the integration of automation concepts through the process of inspection, condition assessment, and deterioration prediction.

The current experience-driven condition rating process requires extensive effort in training and quality control to ensure the consistency of the assigned ratings. Additionally, bridge conditions are recorded by rating scores, which lose the local condition details and the opportunity for supporting well-informed maintenance decisions. The lack of details in the currently available bridge condition database limits the performance of the data-driven models that extract knowledge from past experience to guide decision-making in future maintenance.

Meanwhile, the bridge infrastructure system is historically rich in not only structured tabular data such as the NBI, but also unstructured descriptive data such as inspection reports and maintenance records. The inspection reports generated through the current infrastructure management practices only serve as records of activities, leaving the condition details and domain expertise buried in the reports without being fully exploited for further analysis. To that end, this study identifies visual and textual data from bridge inspection reports as an untapped resource of bridge condition information and mines domain knowledge from a large number of historical inspection reports for automatic condition rating and information extraction.

First, to improve the accuracy and consistency of bridge condition rating, a data-driven automatic condition rating model is proposed that maps natural language descriptions from bridge inspection reports to quantitative condition ratings. A highly interpretable hierarchical attention network employing word and sentence-level recurrent neural network encoders with an attention mechanism was developed to fully exploit the semantics and context of the heterogeneous textual data from the inspection reports. The proposed system was developed using a large collection of

inspection reports collected from the Virginia Department of Transportation database. The developed model outperformed a variety of baseline systems in terms of accuracy and mean error metrics, while a diagnostic investigation of error cases revealed a number of inconsistency issues in the input data. Visualization of the resulting attention patterns demonstrated interpretable insights regarding the mapping of local descriptions to global condition ratings, which can also assist in the rating assignment by highlighting important indicators that may have been overlooked. The application of the proposed system to improve the consistency of bridge condition assessment was demonstrated via two use cases: automated rating recommendation that produces a rating for a given inspection narrative, and data-driven quality control that screens inspector-assigned ratings based on the corresponding narrative descriptions. The quality control application was examined against a series of assumed rating scenarios to illustrate how the proposed framework can reliably detect inconsistent ratings. The proposed framework can serve as a supportive tool for rating recommendation as well as quality control and error case analysis, which can proactively increase the statewide and nationwide consistency of condition rating practices.

Next, to fully exploit the multi-modal data from bridge inspection reports, a deep learning-based fusion approach is proposed for automated bridge condition rating using the visual and textual data from bridge inspection reports. Considering the structure of inspection reports that each contains a collection of images and a sequence of sentences that document local bridge conditions, the proposed fusion approach constructs visual and textual representations from images and sentences separately, and adopts a sequence encoder followed by an attention mechanism to fuse multi-modal representations to support condition rating. While the image-based defect recognition and condition assessment models have been extensively studied in the existing literature, results from this study show that the visual modality alone did not yield satisfactory condition rating performance. Condition rating using textual data from the inspection reports significantly outperformed the visual modality, and the proposed fusion approach introduced further improvements over the uni-modal baselines. This study further investigated the uncertainty of rating predictions under random disturbance introduced by data augmentation and dropout training strategy. The uncertainty analysis showed that 95% of the rating predictions for the testing data vary within 0.535, and referring the uncertain predictions to human investigations can further improve the rating performance. The proposed model can be used to process the bridge condition data collected from the current visual inspection practices to improve rating consistency, and discussions of this study points to the poten-

tial improvement in future inspection data collection that can further facilitate automated condition assessment.

Lastly, an information extraction framework is developed to extract bridge conditions from the inspection reports at a high level of detail. A natural language processing approach was developed to formalize the condition extraction problem by modeling inspection narratives as a combination of words representing defects, their severity and location, and formulating a sequence labeling task that accounts for the context of each word. The proposed framework employs a deep-learning-based approach and incorporates context-aware components including a bi-directional Long Short Term Memory (LSTM) neural network architecture and a Conditional Random Field (CRF) classifier to account for the context of words when assigning labels. Dependency-based word embeddings were also used to represent the raw text while incorporating both semantic and contextual information. The sequence labeling model was trained using bridge inspection reports collected from the Virginia Department of Transportation bridge inspection database and achieved an F1 score of 94.12% during testing. The proposed model also demonstrated improvements compared with baseline sequence labeling models, and was further used to demonstrate the capability of detecting condition changes concerning previous inspection records. Results of this study show that the proposed method can be used to extract and create a condition information database that can further assist in developing data-driven bridge management and condition forecasting models, as well as automated bridge inspection systems.

This dissertation is a collection of three manuscripts that describes the aforementioned research works. Through the presented research outcomes, this dissertation highlights the value of unstructured bridge inspection documentation in supporting automated condition assessment and information extraction for the smart bridge infrastructure system.

Contents

1	Introduction	1
1.1	Background	1
1.2	Identified Limiting Factors	3
1.2.1	Condition Rating Challenge	3
1.2.2	Limited Supporting Data	5
1.3	Motivation	6
1.4	Dissertation Outline and Contribution	8
1.5	Research Challenges	12
2	Literature Review	17
2.1	Natural Language Processing	17
2.2	NLP Applications in Infrastructure Management	19
2.2.1	Information Extraction	19
2.2.2	Text Classification	21
2.3	Visual and Textual Fusion	24
2.3.1	Visual Representations	24
2.3.2	Multi-modal fusion	25
2.3.3	Domain applications	26
3	Mapping Textual Descriptions to Condition Ratings to Assist Bridge Inspection and Condition Assessment Using Hierarchical Attention	28
3.1	Abstract	28
3.2	Introduction	29
3.3	Proposed Research	33
3.4	Research Methodology	37

3.4.1	GRU-based Sequence Encoder	37
3.4.2	Hierarchical Attention	38
3.4.3	Condition Rating	40
3.4.4	Quality Control	42
3.5	Data Collection and Preparation	44
3.6	Results and Discussions	46
3.6.1	Condition Rating Performance	47
3.6.2	Interpretation of Attention Weights	52
3.6.3	Quality Control Performance	58
3.6.4	Analysis of Quality Control Scenarios	59
4	Fusing Visual and Textual Representations from Bridge Inspection Reports for Reliable Automated Condition Assessment	64
4.1	Abstract	64
4.2	Introduction	65
4.3	Proposed Multi-modal Rating Model	68
4.4	Data Collection and Preparation	71
4.5	Results and Discussion	75
4.5.1	Multi-modal Rating Performance	76
4.5.2	Uncertainty Analysis	82
5	Context-aware Sequence Labeling for Condition Information Extraction from Historical Bridge Inspection Reports	87
5.1	Abstract	87
5.2	Introduction	88
5.3	Research Motivation	91
5.3.1	Identified Limitations and Knowledge Gaps	91
5.3.2	Need for Context-aware Information Extraction	93
5.4	Research Approach	94
5.4.1	Word Embedding	95
5.4.2	Bi-directional LSTM	97
5.4.3	Conditional Random Field	98

5.4.4	Bi-directional-LSTM-CRF	99
5.5	Data Collection and Preparation	99
5.5.1	Inspection Report Corpus	99
5.5.2	Ground Truth Labeling Categories	100
5.5.3	Word Embedding	101
5.6	Results and Discussions	103
5.6.1	Model Performance	106
5.6.2	Effect of Context Awareness	108
5.6.3	Analysis of Labeled Inspection Reports	113
6	Conclusion and Future Work	115

List of Tables

1.1	Condition rating for the evaluation of bridge components (<i>e.g.</i> deck, superstructure, and substructure) [1].	4
2.1	Summary of existing information extraction applications in infrastructure maintenance and management in terms of methods, tasks, and topics.	20
2.2	Summary of existing text classification applications in infrastructure maintenance and management in terms of text content, features, and methods.	22
3.1	Two example cases of model-generated probability scores with the inspector-assigned rating of 5.	42
3.2	Statistics of the training and testing sets.	46
3.3	Condition rating performance of the proposed framework compared with baseline systems.	49
3.4	Condition rating performance of the proposed HAN model with different word embeddings.	51
3.5	Condition rating performance with different levels of details in the inspection reports.	53
3.6	List of 20 top-weighted words for different condition ratings.	57
3.7	Quality control performance compared with the baseline system.	58
3.8	Five assumed error scenarios and the approach for generating the incorrect ratings of each error type.	60
4.1	Visual data (images) and textual data (descriptive sentences) for bridge deck from a typical inspection report.	67
4.2	Statistics of the training and testing datasets (Top row: training; Bottom row: testing).	75
4.3	Performance of the proposed fusion model compared with the uni-modal baselines and multi-modal variations.	79

4.4	Impact of the components in the proposed architecture.	81
4.5	Impact of visual feature extractors.	82
5.1	Example condition documentation from Bridge Inspector’s Reference Manual [2]. .	93
5.2	Statistics of one split of the inspection report corpus into training, validation, and testing sets.	104
5.3	Hyper-parameter settings for the modeling pipeline used in this study.	105
5.4	Training, validation and testing F1 score for the proposed Bi-LSTM-CRF model. .	106
5.5	Confusion matrix for all tokens in the ten randomly-split testing sets. (N: Name, L: Location, S(Q): Qualitative severity, S(N): Numerical Severity, O: Other)	107
5.6	Performance of different model architecture variants compared with baseline mod- els (token leve).	109
5.7	Chunk-level and sentence-level performance of the proposed model compared with the two baselines: SVM(w2v) and RNN(w2v).	112

List of Figures

1.1	The pipeline of current bridge infrastructure management system.	2
1.2	Example sentences of condition descriptions and the corresponding image illustrations.	7
3.1	The proposed framework for automatic condition rating and quality control.	36
3.2	The gating mechanism inside a GRU node.	38
3.3	Top: Distribution of bridges from nine VDOT regional offices [3]. Bottom: Characteristics of the collected reports including material, design, and structural deficiency (SD: Structurally Deficient).	45
3.4	Condition rating performance of the proposed hierarchical attention framework.	48
3.5	Training and validation curves with different dropout probabilities.	51
3.6	Histogram of the level of detail (LoD) measure for the testing data.	52
3.7	Example descriptions from the testing set that were correctly assigned a condition rating by the proposed model.	54
3.8	Example incomprehensive descriptions mis-predicted as two levels higher.	55
3.9	Example bridge deck condition descriptions with a condition rating of 6 but mis-predicted as 4.	56
3.10	Quality control performance with different number of data points used for tuning parameter Θ	59
3.11	Quality control performance in five error scenarios.	60
3.12	Variation of F1 scores with the value of Θ in the five error scenarios.	61
3.13	Trade-off between precision and recall (left), and trade-off between the false positive or false negative errors (right) in selecting an optimal value for Θ . (*Figure generated in Rand scenario.)	62

3.14	The optimal Θ values (left) and model accuracy (right) for different percentages of wrong ratings. (*UpTwo and DownTwo cases do not include 0.9 datapoints because not enough ratings can be shifted up or down by two levels.)	63
4.1	Model architecture for the extraction and fusion of the visual and textual representations.	69
4.2	Composition of the collected reports in terms of VDOT districts, number of spans, structural designs, and materials.	72
4.3	Example deck images illustrating local conditions of bridge decks.	73
4.4	Example non-deck images.	74
4.5	Training and validation curves of the proposed multi-modal rating model.	77
4.6	Confusion matrix of the testing predictions	77
4.7	Training and validation curves with different dropout probabilities.	82
4.8	Histogram of model uncertainty given correct/incorrect predictions.	84
4.9	Accuracy gain while considering model uncertainty.	84
4.10	Model prediction uncertainty in different condition categories.	85
4.11	Correlation among predicted softmax scores of different condition categories. . . .	85
5.1	Framework for infrastructure maintenance information retrieval.	90
5.2	The proposed processing pipeline with an example sentence from bridge inspection reports. (N: Name, L: Location, S(Q): Qualitative Severity, S(N): Numerical Severity, O: Other)	95
5.3	Skip-gram model architecture redrawn based on [4]. (w_t : input word; w_{t-2} , w_{t-1} , w_{t+1} : neighboring words; V: vocabulary size; N: word embedding dimension.) . . .	96
5.4	Boxplot of number of words documenting the deck, superstructure and substructure components for over 21,000 bridge inspection reports collected from Virginia Department of Transportation (VDOT).	99
5.5	T-SNE plot of the selected words and the words that were most similar based the word embeddings.	102
5.6	Examples sentences with mispredicted categories.	108
5.7	Comparison of model prediction examples between non-context and context-aware models.	111
5.8	Condition changes between two inspection dates for a 1932 slab bridge in Virginia.	113

5.9	Visual illustration of deterioration progress revealed from historical inspection reports.	114
6.1	Dissertation outline of identified limitations , motivations , and research questions .	116

Chapter 1

Introduction

1.1 Background

The management of the bridge infrastructure system encloses a pipeline of condition data collection (*i.e.* inspections), condition assessment, and deterioration prediction. As presented in Figure 1.1, the current bridge management system relies on experience-driven inspection and condition assessment processes, which have been carefully designed and regulated by domain guidelines and standards, to support deterioration prediction. The National Bridge Inventory Standard (NBIS) requires a routine inspection for most bridges and culverts to be performed at regular intervals not to exceed 24 months [1]. During each inspection, bridge inspection reports are filled out by the inspectors according to the Bridge Inspectors' Reference Manual [2]. Conditions are evaluated by the inspection personnel based on their domain expertise and experience gained from training and practices. Maintenance and management decisions are made based on deterioration models [5–7] and predictive analytics [8–10] that are supported by the condition evaluation results.

The condition ratings obtained from bridge inspections, together with other relative information such as lane closure, performed maintenance actions, are submitted to the National Bridge Inventory (NBI) database [1]. The NBI database is a comprehensive database containing information for more than 617,000 bridges that are longer than 20 feet in the United States [1]. This information includes bridge characteristics (*e.g.* geometry, structural systems, materials, location, etc.), as well as component-level condition ratings assigned through visual inspections that were updated through the years. These ratings are provided for components of a bridge (*e.g.* deck, superstructure, substructure) on a zero to nine deterioration severity scale (zero and nine denote failed bridge and

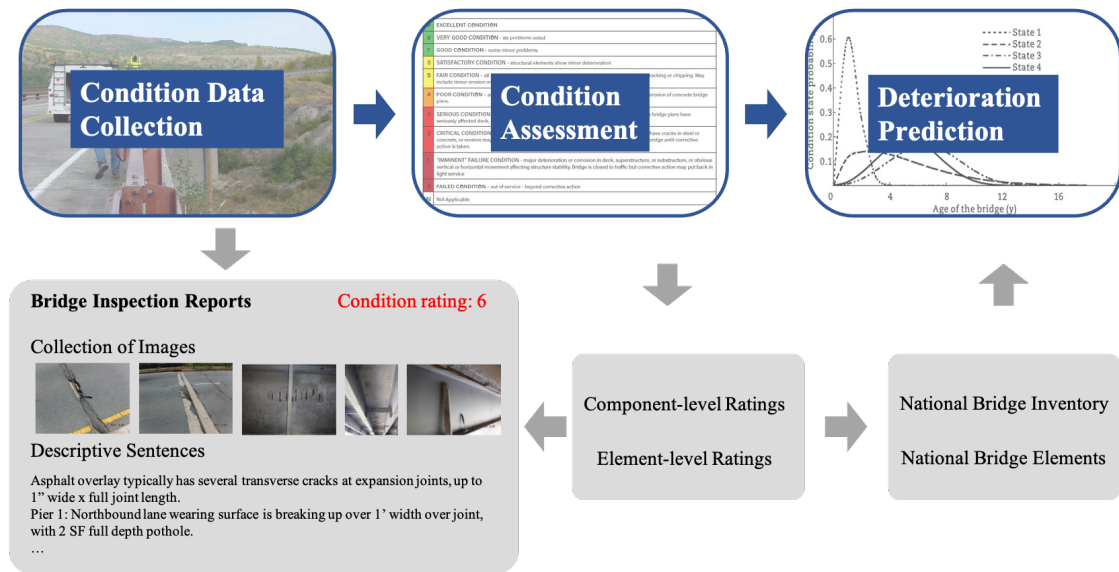


Figure 1.1: The pipeline of current bridge infrastructure management system.

excellent condition, respectively). Having been maintained since 1992, the NBI database provides the basis for resource allocation decisions to ensure the safety and functionality of the nation's bridge infrastructure system.

The Federal Highway Administration (FHWA) and state Departments of Transportation (DOTs) have made substantial procedural efforts to ensure that the condition ratings are properly and consistently assigned by the inspectors. For an inspection team with a typical size of 2-3, most states require more than 5-years of experience as well as prerequisite inspection training programs and state-administered proficiency tests for the lead inspector [11]. Periodic refresher training is also required by the National Bridge Inspection Standard (NBIS) [12] to improve the quality of bridge inspections and maintain consistency throughout the network of inspection programs. Some states also adopt a variety of carefully designed quality control and assurance (QC/QA) procedures to ensure the quality of inspection reports such as the office review process, independent field re-inspection program, an occasional Professional Engineer (PE) ride-along as well as field review [11], and programs that identify inconsistency in infrastructure management data [13–15].

In addition to the component-level ratings in NBI, FHWA also introduced the element-level ratings, which provide a more fine-grained subdivision of bridge members together with quantitative information about the extent of observed deterioration [16]. This standardized system has provided a more comprehensive and objective basis for the documentation, management, and deterioration modeling of assets [17, 18].

While the current experience-driven inspection, assessment, and prediction pipeline have been practices for decades, today's bridge infrastructure system in the U.S. faces critical preservation challenges in the sheer volume of aging and deteriorating bridges with limited funding and resources. For more than 617,000 bridges across the states [19], 42% have reached or passed the design life of 50 years; 7.5% are classified as structurally deficient (in "poor" condition); and over 10% have weight limit posting to restrict heavy traffic loads due to deterioration [20]. One out of every three U.S. bridges has been identified as needing structural repair, rehabilitation, or replacement [21]. The most recent estimation of repair backlog for existing bridges is at \$125 billion. At the current rate of investment (\$14.4 billion annually), it will take until 2071 to complete all repairs that are currently necessary, and the additional deterioration until then will become overwhelming as the rate of deterioration exceeds the rate of repair [22]. Improving the efficiency of the bridge infrastructure system in the face of critical preservation challenges calls for the integration of efficient automation concepts through the process of inspection, condition assessment, and deterioration prediction.

1.2 Identified Limiting Factors

While the efforts by FHWA and state DOTs have established sound inspection and condition assessment procedures, this study identifies two factors in the current experience-driven practices that are limiting the efficiency of bridge management system: First, assigning a condition rating based on experience is a challenging task that requires extensive effort in training and quality control to ensure rating consistency. Second, the current NBI and NBE data simplifies bridge conditions into ratings, which loses the local condition details and the opportunity for supporting detail-driven well-informed maintenance decisions.

1.2.1 Condition Rating Challenge

Assigning condition ratings properly and consistently is a challenging task mainly for the following two reasons. First, the rating guidelines need to be properly interpreted by individual inspectors in order to assign a rating. Second, the component ratings are an overall representation of local deficiencies, which requires bridge inspection expertise to aggregate local conditions to a holistic component-level rating. Regarding the individual interpretation, Table 1.1 presents the descrip-

Table 1.1: Condition rating for the evaluation of bridge components (*e.g.* deck, superstructure, and substructure) [1].

Rating	Guideline
9	EXCELLENT CONDITION
8	VERY GOOD CONDITION - no problems noted.
7	GOOD CONDITION - some minor problems.
6	SATISFACTORY CONDITION - structural elements show some minor deterioration.
5	FAIR CONDITION - all primary structural elements are sound but may have minor section loss, cracking, spalling, or scour.
4	POOR CONDITION - advanced section loss, deterioration, spalling, or scour.
3	SERIOUS CONDITION - loss of section, deterioration, spalling, or scour have seriously affected primary structural components. Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present.
2	CRITICAL CONDITION - advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken.
1	“IMMINENT” FAILURE CONDITION - major deterioration or section loss present in critical structural components, or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put bridge back in light service.
0	FAILED CONDITION - out of service; beyond corrective action.

tive guidelines of fine-grained condition ratings on the scale of 0-9 defined by the FHWA Coding Guideline [1]. As can be seen from the guidelines for each rating, the transition between condition ratings does not involve a crisp boundary and depends on inspectors' knowledge and experience. The Inspectors' Reference Manual [23] also outlines the procedure for determining the proper component condition ratings following the guideline, which requires that the inspector read through the descriptions in their notes taken from field inspection and carefully examine the rating scales from condition 9 to condition 0 until encountering a condition rating whose guideline description is more severe than those in their notes. The proper condition rating resulting from this process should be one level higher than the identified more severe condition rating. The state Departments of Transportation (DOTs) have also developed supplemental component condition rating guidelines [24, 25] to lay out their specific procedure in compliance with these guidelines and further aid the inspectors in assigning the ratings. Concerning the overall representation of local deficiencies, the FHWA Coding Guideline states that the component condition ratings should reflect the overall conditions of the component rather than localized conditions [1]. This also makes the assignment of an overall condition rating based on individual member conditions challenging. The inspectors rely on their bridge engineering expertise to determine the impact of local deficiencies on the overall component condition when assigning component condition ratings. Given the above-mentioned reasons, assigning condition ratings consistently and accurately is a challenging task, and innovation is required in the development of intelligent methods of rating estimation and quality control. To the above regards, it requires significant time, budget, and human efforts for inspector training and additional caution to ensure the quality of the condition rating data.

1.2.2 Limited Supporting Data

In addition to the challenge for assigning the ratings, the sparse condition data in NBI and NBE presents a limitation in supporting system-level infrastructure analytics and smart management decision-making. While the NBI database provides the basis for resource allocation decisions, the coarse rating system in NBI has major shortcomings from the following aspects: spatial resolution, defect categorization, and defect quantification. Condition ratings in NBI are evaluated and assigned only for bridge components such as decks, superstructures and substructures, rather than individual structural elements (girders, piers, etc.). The ratings include a single number that is an aggregate of the overall condition of the component and thus indicate no information about

the specific defect types detected (*e.g.* cracks, corrosion, spalling, etc.). This system provides no information about the extent or size of specific defects (*e.g.* crack width, corrosion area, etc.), and no information is provided on the location of the defects on the structural members, which makes it impossible to track the progression of specific defects over the life cycle of the structure. While the element-level ratings in NBE provide important improvements in terms of spatial resolution, defect categorization, and quantification, a more detailed level of condition information such as measurements and location of individual defect features is still not captured. Such information is essential in tracking the evolution of a specific damage feature between multiple consecutive inspections and in modeling the time history evolution of damage with the aim of damage prognosis and remaining life estimation [26, 27].

1.3 Motivation

The bridge infrastructure system is historically rich in not only structured tabular data such as the NBI and NBE, but also unstructured descriptive data such as inspection reports and maintenance records. These descriptive data are historically consistent, rich in details, and encoded with engineering expertise. For the example of bridge inspection reports, having been maintained by the state Departments of Transportation (DOTs) for decades, bridge inspection reports record the condition evolution of the entire bridge population. During each inspection, bridge inspection reports are filled out by the inspectors according to the guidelines provided by Bridge Inspectors' Reference Manual [2]. The inspection reports contain narrative descriptions as well as illustrative images of local deficiencies as discovered during an inspection (as presented in Figure 1.2). The descriptions are in the format of natural language, where each sentence describes a finding by the inspector that typically contains the name of local deficiency, its locations, and its severity and extent. Following each inspection, condition ratings were assigned by the inspection personnel for bridge components such as deck, superstructure, and substructure on a scale of 0-9.

On the one hand, these condition ratings collected over the years, together with the visual and textual data from the inspection reports, represent a knowledge base that aggregates individual inspectors' expertise of bridge condition assessment, and can shed light on how local deficiencies map to global condition ratings. This motivates the development of an automated condition rating model that learns from the collective expertise and knowledge embedded in massive historical reports to promote a unified and consistent approach for assigning condition ratings. Central to this

Joints are deteriorated and leaking full length throughout and with minor vegetation growing from joints at abutments.



Area of delamination and spalling, 20' long x full height x up to 3" deep with up to 25% section loss to exposed reinforcing.



Figure 1.2: Example sentences of condition descriptions and the corresponding image illustrations.

hypothesis is the assumption that the raw narrative descriptions and images are more objective and subject to less variability from an inspector's experience-driven judgment as opposed to the ratings. Unlike the condition ratings which require substantial training and experience to be properly and consistently assigned, describing and taking images of bridge condition observations as seen during an inspection is expected to involve less subjectivity and variability.

On the other hand, the descriptions in the inspection reports contain information regarding local deficiency details and their evolutionary history. Information at such a detail level is not currently available in the NBI and NBE data. The lack of automation in information extraction at such levels of detail is a barrier to developing high-accuracy deterioration modeling systems and effective maintenance planning and resource allocation decisions. For instance, a maintenance manager's access to such automation solutions can provide timely answers to *maintenance planning queries* such as "How many and which bridges have experienced major crack growth since the last inspection?" and "What is the projected rate of deterioration for those bridges in the next 10 years?". These capabilities will translate into more informed decisions and more effective maintenance of bridge inventories.

The inspection reports generated through the current infrastructure management practices only serve as records of activities, leaving the condition details and domain expertise buried in the reports without not fully exploited for further analysis. This untapped resource of bridge condition information is valuable in its large-scale, rich details and unique domain expertise, and offers the opportunity for supporting automated condition rating and information extraction models.

Besides the visual data from bridge inspection reports, a significant number of robotic inspection systems developed recently are also producing an increasing volume of data that can potentially

support the automatic condition rating models. Robotic inspection systems adopt unmanned aerial or ground vehicles (UAV/UGV) [28, 29] to navigate around bridges and collect visual data [30–33]. As of 2018, the Department of Transportation (DOT) in 15 states are actively conducting research in the use of UAVs in field inspections; 20 states have incorporated UAVs into their daily operations [34]. The robotic inspection systems demonstrate great potential in alleviating labor cost and safety concerns in visual inspection, and more importantly, it produces increasing data support for automation in the subsequent condition assessment process.

1.4 Dissertation Outline and Contribution

The overarching goal of this dissertation is to mine domain knowledge from unstructured multi-modal data from bridge inspection reports to advance automation in smart bridge infrastructure management. This dissertation identifies visual and textual data from bridge inspection reports as an untapped resource for bridge condition information and mines domain knowledge from a large number of historical inspection reports using deep learning-based natural language processing as well as visual and textual fusion techniques. Automated condition rating models are developed using textual and visual data from the inspection report. A quality control tool is developed based on the rating model to proactively improve the consistency of inspector-assigned ratings. Uncertainty of the condition rating model is also evaluated using an approximate Bayesian neural network setting. Local condition information is extracted from the unstructured textual description in the inspection reports via a context-aware sequence labeling task that segments and structures sentences into categorized chunks of defects, locations, and severity. To the above regards, three research efforts were carried out, and the outcomes will be presented in the following chapters in the format of a collection of three manuscripts currently in different stages of the peer review process in research journals.

Chapters 3 and 4 focus on the research question of automated condition rating. A data-driven framework is proposed in chapter 3 to map natural language descriptions to quantitative ratings in order to improve the accuracy and consistency of these ratings. A hierarchical architecture employing recurrent neural network encoders with an attention mechanism was developed that progressively summarizes from the word-level to the sentence-level and ultimately to the document-level representations to support automatic condition rating. Visualization of the resulting attention patterns was shown to provide interpretable insights which highlight potentially overlooked indicators.

Application of the system via two use cases further showed the potential of the system as a supportive tool for automated rating recommendation as well as quality control, which can proactively increase the consistency of condition rating practices. This study contributes to the knowledge of infrastructure management in the following aspects:

- This study is the first to collect textual data from a large amount of historical bridge inspection reports and construct a data-driven approach that maps narrative descriptions to condition ratings. The textual data contains rich details of historical bridge conditions in terms of local deficiencies as well as their severity and locations. The large number of reports and their associated ratings represent a knowledge base that aggregates the expertise of various bridge inspectors, and hence can be used to support a data-driven approach to produce consistent condition ratings.
- The proposed framework develops a deep-learning-based approach to fully exploit the semantics and context of the highly heterogeneous textual data from bridge inspection reports, and studies the problem of assigning a rating score instead of a typical categorization of content types. The model is shown to outperform a variety of existing baseline approaches in the civil infrastructure domain literature.
- The hierarchical attention architecture used in the proposed framework is highly interpretable. By identifying the words and phrases that received stronger attention scores, the proposed model can help reveal the latent trends and logic of how the bridges were rated by the inspectors. The resulting attention maps can help inspectors identify features that drive the model-based rating recommendations which may have been overlooked by the inspector, thus contributing to rating quality improvement.
- The proposed framework develops a quality control process that evaluates the condition ratings assigned by inspectors during inspections based on the narrative descriptions in their inspection notes and issues alerts when the assigned rating is inconsistent with the data-driven model output. Such a supportive tool allows for quality control and error case analysis, which can proactively increase the statewide and nationwide consistency of condition rating practices.

Aiming to construct an automated condition rating model that exploits domain expertise in the

multi-modal data from bridge inspection reports, Chapter 4 contributes to the knowledge of smart infrastructure management in the following aspects:

- While the bridge inspection reports contain both visual and textual data, the sufficiency of each modality in supporting automated condition rating has not been compared. This study constructs uni-modal baselines using visual and textual data separately to evaluate and compare their performance.
- Given the structure of inspection reports that each contains a collection of images and a sequence of sentences, where each image or sentence corresponds to a local bridge condition but without explicit one-to-one alignment, this study proposes a deep learning-based fusion approach that extracts the visual and textual representations separately, and adopts a bi-directional RNN sequence encoder followed by an attention mechanism to fuse multi-modal representations.
- This study further develops an evaluation of the model uncertainty incorporated by random image data augmentation transforms and the dropout strategy using repeated testing experiments. The standard deviation of the resulting distribution of model predictions was used as the quantification of uncertainty, and demonstrated reliable rating predictions with relatively small variations.
- Analysis of the resulting predictions revealed the rating behavior under disturbance, and pointed to an use case of model uncertainty, where high uncertainty model predictions can be referred to human inspectors for further investigation. Filtering the model predictions with uncertainty was shown to improve the rating performance, and provides a viable approach for reliably adopting automated condition assessment tools via cyber-human collaboration.

Lastly, Chapter 5 focuses on the Information Extraction (IE) problem and builds a model to extract structural condition information from bridge inspection reports in order to assist data-driven bridge maintenance and management. This chapter models inspection narratives as a combination of defect names (N), their location (L), and severity (S) that are arranged in the heterogeneous and complex patterns of natural language to describe an infinite variety of real-life inspection scenarios. This study adopts deep-learning-based information extraction framework in the field of bridge infrastructure maintenance and management to perform context-aware information extraction. The proposed approach presents contributions in primarily the following aspects:

- This research identifies textual data in the bridge inspection reports as an untapped resource for bridge condition information and introduces a natural language processing framework to extract condition information. The extracted information forms a bridge condition inventory that can further support data-driven maintenance analytics and decisions.
- Information extraction was formulated as a context-aware sequence labeling task, where the label of a word is assigned based not only on the current word, but also its semantic and contextual relations in the sentence. This is especially important in the field of bridge inspections because many technical details such as measurements and structural member names can be interpreted differently based on their context. For example, the words "12 inch" describe a defect location in phrase "12 inches above pier 2", but "(a crack) 12 inches long" describes the severity of the defect. The method proposed in this paper is capable of accounting for context and semantic relations, which is essential for efficient information extraction from bridge inspection reports.
- The context-aware sequence labeling model also provides the capability to output chunks of condition-related information instead of broken pieces of individually labeled words. The labels generated by the proposed method are designed to be more continuous so as to chunk the sentences into condition-related phrases. This continuity is enforced through the adoption of context-aware components including the Conditional Random Fields classifier, as well as the sequence-driven formulation of bi-directional Long Short-Term Memory neural network, and is motivated by the desired use case of information extraction from historical inspection reports with the aim of assisting big bridge data analytics.
- Finally, this study puts the formalized engineering knowledge into the context of bridge maintenance and management by providing example use cases of how the proposed method can be used to support maintenance planning queries and tracking the changes in consecutive inspection reports. The success of the proposed research is expected to provide a feasible approach to extract meaningful information from textual bridge condition records, and can potentially be applied to across the 50 states to form a knowledge base for detailed bridge condition history. This condition inventory can support bridge owners and management agencies in their large-scale maintenance planning and decision-making by providing targeted insights and historical condition comparison.

1.5 Research Challenges

This section lists the research challenges and shortcomings highlighted by the extensive review of existing visual and textual information extraction works in infrastructure management system literature. The detailed literature review will be presented in the next chapter.

Challenge 1 Textual data from bridge inspection reports are heterogeneous and complex since it comes from various inspectors and contains a mixture of technical and non-technical words. Being able to exploit the context and semantics behind the words is essential for successful condition rating and information extraction from bridge inspection reports. Unlike other domain textual data such as building codes and energy conservation codes that are regulatory documents in formal written language composed by a group of experts, bridge inspection reports are created by various professional inspectors from across the states to document their findings during field inspections, with no standard requirements on the type of language and wording to use in the narratives, and hence can contain higher variance in word usage.

The existing rule-based and ML-based NLP applications [35–38] in the domain of civil infrastructure management have limitations in addressing the challenges associated with the heterogeneous nature of textual data from bridge inspection reports. While the existing rule-based approaches can be tuned to achieve satisfactory performance for a very specific task, it is difficult to define a set of all-encompassing rules that cover every variant scenario. The ML-based models usually leverage syntactical features [39, 40] or word frequency as features [38, 41, 42], which limits the model’s ability to exploit the semantics contained in the textual data. Ontology-based semantic features [39, 43] have been adopted for information extraction from bridge inspection reports, but the ontology is able to cover only a selection of words, and the quality of the semantic features relies heavily on the design of the categories and the comprehensiveness of the ontology. The use of context-aware deep-learning-based methods is scarce in the field of infrastructure management and maintenance, while the complexity of the textual data from infrastructure inspection reports demands such a model to capture the various usages of words and extract correct information.

Challenge 2 Bridge inspection reports contain document-level textual data that consists of sentences of words. Depending on the number of local deficiencies that were identified by the inspection personnel and the length of the description provided for each deficiency, the length of an

inspection report varies and a long report can contain even 4-5 thousand words. To capture the nuances of context and semantics in such long documents, and summarize these into an overall condition assessment is a challenging task.

The frequency-based features such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) vectorize a document by calculating and weight the frequency of each word. Although such features reveal the selection and distribution of words and hence provide the basis for the categorization of sentences or documents, the sequences of words were reduced to a sparse matrix of frequencies, without capturing the nuances in the order of words and their semantics. Transformer-based models have demonstrated good transferability with large-scale multi-task model pre-training and light fine-tuning for state-of-the-art performance in downstream tasks. However, the significant computation resources required by the Transformer-based model limit its capability in processing long text sequences, and the performance of transferring model pre-trained on general texts (*e.g.* Wikipedia) to domain-specific texts such as inspection reports is still unknown.

Challenge 3 The existing automatic condition assessment works, mostly image-based [44–48], are focused on the detection of local deficiencies such as crack [49, 50], delamination [51], and spalling [52] from various surface materials such as steel [53], and asphalt and concrete [54–56]. However, it requires further research efforts to progress towards damage quantification, aggregating local deficiencies to global condition ratings, and linking these assessment results to subsequent maintenance decision-making.

Challenge 4 While visual and textual bridge condition data have become increasingly available, the problem of fusing both modalities for automated condition assessment has not been extensively researched. A large amount of research has developed models for visual and textual fusion using large-scale, general-content data [57–60], while their applicability to the domain of bridge condition assessment remains unknown due to the unique content and structure of bridge inspection report data. The descriptive sentences state in detail every local defect identified during inspections, while the images serve as supplementary information that illustrates zoomed views of selected local defects. Each image and sentence correspond to one or more inspection findings, typically local defect conditions, without direct alignment as image-sentence pairs. The existing fusion approaches were developed for input modalities that have perfect alignment and contain complementary information [59]; however, fusing such loosely-aligned modalities with supplementary information such

as the visual and textual data from the inspection reports is still challenging and warrants further study.

Challenge 5 Typical information of interest when evaluating the condition of a bridge includes the local deficiency (type), its measurements (severity), and where it is located (location). Each piece of information functions only when put into the proper context. The Bridge Inspector’s Reference Manual [2] requires that when documenting a deficiency encountered during inspections, the exact location, severity, and extent of deficiencies should be specified to determine the bridge condition. Such information comes in chunks instead of single words when quantifying or localizing bridge deficiency conditions. Therefore, extracting chunks of information instead of individual words is especially important for bridge inspection. For example, instead of extracting the word ‘2’ as a number followed by the word ‘feet’ as a unit, it is important to know when ‘2 feet’ is used to describe the location of spall and when it is used to describe the size of a spall, since the two situations raise different levels of concerns in bridge inspection.

In the sequence labeling task to extract chunks of information, the labels should be assigned based on their context such that the same word can be assigned different labels when in different contexts that refer to different categories of condition information. In other words, the desired sequence labeling model should not memorize every word and assign a label in a dictionary-lookup manner, but should predict the label based on the meaning of the word and its context in the sentence. This is not only because building an all-encompassing dictionary to look up labels is obviously cumbersome, but the benefits of a context-aware labeling system also lie in the flexibility of allowing the same word to be labeled differently in different contexts. To that end, the same word should be allowed to have different labels in different contexts, and the desired labels help group the words in a sentence into chunks of conditions (*e.g.* name, location or severity), instead of trying to classify each word independently into an entity category. This arrangement is rooted in both the way that bridge inspection information is documented, and the desired use case of the extracted information from the inspection reports. While the existing sequence labeling model considers a context window of only one [39], the desired feature of target information requires a sequence labeling model that can recognize the semantics and dependencies within the context among the input sequence of words.

Challenge 6 Current routine bridge inspection is performed annually or biannually, and the condition assessment from the bridge inspection is used to support maintenance decisions. Although this condition assessment is critical to bridge maintenance, rarely is there an opportunity to investigate the reliability of the bridge inspection practice and quantify the uncertainty in the condition assessment, as the basis for comparing the assessment (*i.e.* the bridge condition) varies from year to year.

Considering the significance of the current bridge inspection and condition rating practices, Federal Highway Administration (FHWA) has conducted a reliability study of visual inspection and condition assessment in collaboration with the nondestructive evaluation validation center (NDEVC) in the year 2001. In the study, seven of the NDEVC test bridges were used to define 10 discrete inspection tasks. Inspectors from 25 states participated in this study, which can be considered as sampled from the entire practicing bridge inspectors. This study identified various factors that can influence the accuracy and comprehensiveness of visual inspections, including subject factors (*e.g.* visual acuity, color vision, experience), physical and environmental factors (*e.g.* lighting, viewing aids, background noise), task factors (*e.g.* inspection time, the spatial distribution of items) and organization factors (*e.g.* training, standards). Results from the reliability study show that 58% of the individual ratings were assigned incorrectly by the participating inspectors when compared with the rating of the NDEVC reference. Using the sample from this study, it is estimated that 95% of the condition ratings for the entire bridge population vary within approximately two rating points from average; 68% of the ratings vary within one rating points. Also, “better” condition elements tend to be rated lower, and “poorer” condition elements were rated higher than reference ratings; more errors were found in rating “poorer” condition elements, which is undesired for a rating system since “poorer” condition elements are the ones that play critical roles in maintenance decision making. It should also be noted that in the reliability study, participants of practicing bridge inspectors were tasked with inspection procedures that were slightly different from their daily practices, such as unfamiliar bridges and smaller or no inspection team. The current bridge inspection and condition assessment is considered as sound practices, evidenced by years of safe operations of the bridge infrastructure and the regulatory efforts made by the bridge management agencies and practitioners to ensure the quality of the condition ratings. Nevertheless, the results of the reliability study showed significant variability in the visual inspection and condition ratings, which calls for the development of mechanisms that reveal and reduce the uncertainty of the ratings for more reliable bridge

condition assessment that supports maintenance decision-making. This challenge also applies to the automatic models developed for defect detection [61, 62], defect quantification [45, 63], and robotic inspection [30–33]. Reliability is usually the major concern of bridge management agencies when considering adopting automatic models in their daily practices, which calls for the consideration of model uncertainty in developing automatic approaches.

Chapter 2

Literature Review

2.1 Natural Language Processing

Natural language processing (NLP) develops models and techniques that automatically analyze data in the form of natural languages such as text and speech. Major applications of NLP include information extraction, sentiment analysis, machine translation, and question answering. The NLP models developed for these applications can be categorized into two main groups: rule-based methods [64, 65] and machine learning-based (ML-based) models [66, 67]. The rule-based models require extensive human efforts in designing rules of syntactical pattern matching for successful information extraction [68]. Machine learning models learn the rules and recurring patterns from large datasets using features extracted from raw textual inputs. Features that have been exploited by machine learning-based approaches include spelling features such as word cases, prefixes or suffixes; contextual features such as context words, part-of-speech labels; ontology-based features (semantic word relations extracted from pre-defined dictionaries); and dense vector representations of words. The dense vector representation (word embeddings) is mainly used to support deep learning models and has proven to excel in multiple benchmark tasks without engineered task-specific features [69], and hence has received extensive research attention during the past decade [70, 71].

Sequence labeling Many NLP applications can be formulated as sequence labeling problems, where sentences are processed so that each word is assigned a pre-defined label. Sequence labeling models can achieve different goals depending on how the labels are defined. Toutanova et al. [72] trained a maximum entropy classifier that assigned part-of-speech tags such as tense, number (plural/singular), or case using a rich set of features. Kudoh and Matsumoto [73] developed a se-

quence labeling model using Support Vector Machines that assign labels to segment the sentence into syntactically-related phrases (*e.g.* noun phrase, verb phrase). Collobert et al. [69] proposed to avoid task-specific feature engineering by starting from randomly initialized dense word embeddings and optimizing the embeddings during the training of deep learning models. This approach has proven to excel in multiple NLP benchmark tasks. Learning word embeddings from a large corpus and building deep learning models upon these pre-trained word embeddings has demonstrated state-of-the-art performance in sequential labeling benchmark tasks [74, 75]. Recent studies proposed incorporating character-level embedding representations to further improve the performance [76–78]. Deep-learning-based models have shown promising results in sequence tagging tasks using both convolutional neural network models [79, 80] as well as recurrent neural networks [81–83]. Recurrent Neural Network (RNN) and its cell-mechanism variants, Long-Short-Term Memory (LSTM) [84] and Gated Recurrent Unit (GRU) [85], has been widely adopted to process sequential inputs [86, 87].

Text Classification Text classification processes textual data (sentences or documents) and categorizes them into labels of interest. A body of Machine Learning (ML) models has been developed for text classification such as Logistic Regression (LR) [88] and Support Vector Machine (SVM) [89, 90]. Frequency-based features such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) were developed to support the ML models, where the sentence of documents is vectorized by calculating and weight the frequency of each word. Although the frequency-based features reveal the selection and distribution of words and hence provide the basis for the categorization of sentences or documents, the sequences of words were reduced to a sparse matrix of frequencies, without capturing the nuances in the order of words and their semantics. To that end, various dense vector representations have been developed to better represent contextual and semantic information. Such dense vector representations, *e.g.* Word2Vec [70], GloVe [71], and fastText embeddings [91], were usually trained using large text corpora to be encoded with general word co-occurrence statistics, which links to the semantics of words based on the linguistic theory that the meaning of each word lies in its context and co-occurrence with neighboring words [92]. Building on the idea of representing words with dense vectors, Deep Learning (DL) models have been developed for classifying textual inputs, including Convolutional Neural Network (CNN) [93–95], Recurrent Neural Network (RNN) [96, 97], and Transformer-based models [98, 99]. The CNN take sequences of dense word vectors as input, and model the dependencies among input se-

quences using convolution over neighboring words; while the RNN process sequential input vectors one-by-one and model both neighboring and longer-term dependencies with recurrent connections among RNN cells. Transformer-based models take sequences of dense word vectors and the corresponding sequences of word order indices as input, and adopt multi-head self-attention modules to model the complex relationships between each pair of input words. Transformer-based models are more suitable for parallel computation than RNN, and have demonstrated good transferability with large-scale multi-task model pre-training and light fine-tuning for state-of-the-art performance in downstream tasks. However, the significant computation resources required by the Transformer-based model limit its capability in processing long text sequences, and the performance of transferring model pre-trained on general texts (*e.g.* Wikipedia) to domain-specific texts such as inspection reports is still unknown.

2.2 NLP Applications in Infrastructure Management

2.2.1 Information Extraction

In the field of infrastructure maintenance and management, NLP applications have been developed to extract engineering information from domain-specific textual data. While these methods have been developed for various applications and topics of textual data in the general domain of infrastructure maintenance and management, they can be broadly categorized into rule-based, machine-learning-based (ML-based), and deep-learning-based (DL-based). Table 2.1 summarizes these applications in terms of methods, tasks, and the topic of their textual data. The rule-based methods rely on human efforts in designing rules of syntactical pattern matching for successful information extraction. Abuzir and Abuzir [68] developed a system that relied on human-defined syntactical patterns to identify civil engineering terms and the relations among them using text from web pages. The extracted terms and relations were used to construct a thesaurus in the field of civil engineering for supporting automatic information retrieval systems. Al Qady and Kandil [100] applied a rule-based shallow parser to identify concepts and their relations from construction contract documents to assist efficient project management and contract administration. El-Gohary and El-Diraby [101] proposed to construct a domain-specific ontology for domain-wide integrated construction and infrastructure development using experts' knowledge of domain-specific concepts and relation structures. Zhang and El-Gohary [35] built a rule-based pattern matching system that uses syntactic

Table 2.1: Summary of existing information extraction applications in infrastructure maintenance and management in terms of methods, tasks, and topics.

Method	Task	Reference	Topic
Rule-based	Ontology Construction	[68]	Civil Engineering
	Entity and Relation Extraction	[100]	Construction
	Ontology construction	[101]	Construction
	Entity Extraction	[35]	Construction
	Ontology construction	[43]	Bridge
Rule- & ML-based	Concept & Relation Extraction	[36]	Construction
	Concept Extraction	[37]	Transportation
ML-based	Knowledge Discovery	[38]	Marine Structure
	Entity Extraction	[39]	Bridge
	Dependency relation extraction	[40]	Bridge
	Named Entity Normalization	[42]	Bridge
DL-based	Concept & Relation Extraction	[102]	Transportation
	Sequence Labeling	this work	Bridge

features and ontology to extract information from construction regulatory documents such as building codes and energy conservation codes. Liu and El-Gohary [43] proposed a bridge data analytics framework and presented an ontology of bridge deterioration knowledge. The taxonomies were manually constructed based on a review of relevant manuals and guidelines.

ML-based applications learn the rules and patterns from large datasets using features extracted from raw textual inputs to extract targeted information. Lee et al. [38] proposed a knowledge discovery system from the inspection reports of marine structures. The system integrated components including a keyword-based document retrieval, document clustering and classification, and trend pattern analysis. Liu and El-Gohary [39] proposed a machine learning-based sequence labeling model that extracts categorized entities from bridge inspection reports using a domain ontology. The ontology contained 11 categories of concepts and terms extracted from relevant manuals and guidelines such as bridge elements, deficiencies, and so on. As such, the resulting semantic features for a certain word were 11-dimensional binary features demonstrating whether the word belongs to each category in the ontology. Liu and El-Gohary [40] developed a dependency parsing

method to extract relations among entities from bridge inspection reports and the preliminary study achieved an accuracy of 78.0%. Liu and El-Gohary [42] proposed an unsupervised Named Entity Normalization method to reduce the discrepancy in the extracted terms from bridge inspection reports. Integrated systems that combine rule-based methods with machine learning models have also been proposed in the literature. Kim and Chi [37] constructed an information retrieval system that extracts construction accident cases from accident records using both rule-based and ML-based method and compared their performances. Zhang and El-Gohary [36] proposed an integrated platform that used a rule-based model to extract Building Information Modeling (BIM) concepts from compliance regulatory documents and a ML-based method to classify their relationships.

A few DL-based NLP models have been developed for infrastructure maintenance and management applications, which further exploit the semantics of textual data via word embeddings and deep neural networks. Le and Jeong [102] developed a semantic classification framework that identifies domain-specific entities and terminologies from heterogeneous transportation asset documents and determines the semantic relations among the terms. Noun phrases were extracted from highway guidelines and manuals, from which domain-specific entities and terminologies were selected. Neural Network models were trained to generate word vector representations of the selected terminologies and semantic relations such as similar-to (synonymy), is-a (hyponymy), and part-of (meronymy) were classified based on the vector representations.

2.2.2 Text Classification

Text classification applications have been developed to categorize raw domain textual data. Table 2.2 summarizes these applications in terms of the text contents, features, and methods. A rich body of research is available in the literature constructing text classification models using word-frequency-based features and ML-based models for domain-specific tasks. For example, Caldas et al. [103] developed classification models to categorize construction project documents such as specifications, meeting minutes, and field reports into 13 pre-defined categories (*e.g.* schedule, HVAC, fire protection). Using TF-IDF as features, different ML-based models were tested and compared including Naïve Bayes (NB), Decision Trees (DT), k-Nearest Neighbors algorithm (k-NN), and Support Vector Machine (SVM), and achieved accuracies ranging from 49.11% to 91.12%. Al Qady and Kandil [104] adopted an unsupervised clustering approach to categorize seventy-seven construction project documents based on their textual similarity. The documents were first clustered

Table 2.2: Summary of existing text classification applications in infrastructure maintenance and management in terms of text content, features, and methods.

Group	Content	Feature	Method	Reference
ML-based	Construction project documents	TF-IDF	Multiple	[103]
	Construction project documents		Clustering	[104]
	Construction hazard scenarios		SVM	[105]
	Construction project documents		NB, k-NN	[106]
	Construction contracts		Multiple	[107]
	Environmental regulations		Multiple	[108]
	Construction accident records		Ensemble	[109]
	BIM case documents		SVM	[110]
DL-based	Power grid malfunction reports	One-hot embedding	LSTM	[111]
	Citizen request sentences	Word embedding	CNN	[112]
	Construction hazard sentences	Word embedding	BERT	[113]

based on their TF-IDF features, and the clusters were then refined by thresholding the cosine similarity between each document and the cluster centroid. Chi et al. [105] used the TF-IDF features and an SVM classifier to categorize descriptions of construction work scenarios into different activity and hazard types to assist Job Hazard Analysis (JHA), which identifies potential task-related hazards and provides safety solutions. Al Qady and Kandil [106] tested different ML-based methods such as NB and k-NN built upon TF-IDF features to categorize construction project documents that provide the factual background of each construction claim into eight corresponding claims. Salama and El-Gohary [107] utilized TF-IDF features and ML models including NB, Maximum Entropy (ME), and SVM to classify 330 clauses (each contains several words) obtained from construction contracts into 14 pre-defined categories such as environmental, safety, health, etc. To select the most informative TF-IDF features, several feature scoring methods were tested including Information Gain (IG), Mutual information (MI), and Chi-squared (χ^2). Zhou and El-Gohary [108] developed a hierarchy of topics for environmental regulatory documents and performed text classification on clauses from the documents to assist environmental compliance checking. Different word-frequency-based features were tested including TF-IDF as well as its supervised variants TFRF and $TF_{max}RF$, where the RF stands for relevance frequency that measures term relevance to the label categories. Various

ML-based methods such as NB, SVM, and k-NN were compared and the best-performance model achieved 97% recall and 84% precision. Zhang et al. [109] developed an ensemble model based on voting among five different ML-based methods built upon TF-IDF features and achieved 68% F1 score to categorize records of construction site accidents into pre-defined accident causes. Jung and Lee [110] compared an unsupervised similarity-based approach with a supervised SVM classifier and achieved 80.75% F1 score for categorizing Building Information Modeling (BIM) case documentation by their types of BIM usage. For unsupervised categorization, the similarity between each document and the definition of a particular BIM use were computed based on their word frequency features, and documents with similarity above a threshold were categorized as the type of BIM use.

Deep-learning-based text classification systems have also been developed for the infrastructure domain. Wei et al. [111] constructed a Long Short-Term Memory (LSTM) Model that classifies the fault categories for power grid malfunction reports. The LSTM model was built using a sparse word vector representation of one-hot embedding and the optimal hyper-parameter setting achieved 64.59% accuracy in fault categorization. Kim and Hong [112] adopted a CNN-based sentence classification method to categorize short paragraphs of transportation-related citizen requests into 14 pre-defined categories. Fang et al. [113] adopted a Bidirectional Transformers for Language Understanding (BERT) model to classify single-sentence descriptions of potential hazards in construction sites into 170 hazard categories and achieved an accuracy of 86.91% during testing.

While these applications contributed to the advancement of automation in construction and infrastructure management, the gaps in the current applied models can be summarized into the following three aspects. First, the majority of the reviewed studies used word-frequency-based features such as TF-IDF, TFRF, and $TF_{max}RF$ to support text classification models, which capture limited semantic information from the textual data. Given the linguistic theory that semantic information lies in the co-occurrence of words [92], the frequency-based features capture a global word co-occurrence pattern in the sentence or document to be classified, and hence contain a certain level of semantic information. However, transforming a sentence or document into word frequencies results in the loss of the local contextual information of each word. Few studies, except [112, 113], used dense word embeddings that are encoded semantic and contextual information. Second, the use of context-aware deep-learning-based methods is scarce in the field of infrastructure management and maintenance, even though the complexity of the textual data from infrastructure inspection reports

demands such models to capture the various usages of words and extract correct information. This need is highlighted by the highly heterogeneous nature of the inspection reports. Infrastructure inspection reports are created by various professional inspectors with no standard requirements on the type of language and wording to use in the narratives, and contain a mixture of technical and non-technical words. Being able to model the semantics and contexts of the words is essential for supporting the successful analysis of complicated inspection reports. Third, existing classification applications focus on categorizing textual data from dissimilar types, *e.g.* Caldas et al. [103] categorized construction project documents into types such as scheduling, HVAC, and fire protection. However, the existing applications did not focus on the problem of assigning a rating score to texts of the same type. Textual data with contents of different types tend to have different sets of words commonly used in each type, which can be leveraged in training a classification model. However, narrative descriptions from bridge inspection reports contain a somewhat similar set of words describing bridges and their conditions. Assigning a rating score requires a model to have a better understanding of the semantics of the textual inputs in order to support the inference of a rating score. Whether a classification model can differentiate between descriptions of different severity of bridge deficiencies to support automatic rating is still unknown and warrants investigation.

Among the applications reviewed above, five ML systems and two DL alternatives were selected to be used as baselines for performance comparison. The ML baselines include Naïve Bayes (NB), Decision Trees (DT), k-Nearest Neighbors (k-NN), logistic regression (LR), and Support Vector Machines (SVM), which use TF-IDF as features. Furthermore, the LSTM-based work by Wei et al. [111] focuses on document classification and is therefore comparable to the work presented in this paper. As a result, it was selected as a DL-based baseline in addition to a GRU-based alternative [114]. In the absence of prior work on classifying bridge inspection reports, these models were implemented and trained on the dataset collected herein to allow for comparison with the results in the present study.

2.3 Visual and Textual Fusion

2.3.1 Visual Representations

Convolutional Neural Networks (CNNs) represent the state of art in image classification tasks that categorize each image with a pre-defined label. The convolution operation is powerful in extracting

local visual feature representation, and the deep layers of CNN contribute by integrating visual features from different levels of abstraction [115]. CNN-based image classification models [116–118] have been developed using the large-scale ImageNet dataset [119] and achieved state-of-art performance in the past decade. The intermediate visual representations learned by a pre-trained CNN model are shown in recent works to be transferable to other visual recognition tasks [120–122]. Considering that a bridge inspection report has a collection of images corresponding to one condition rating, this multi-to-one correspondence relates to the problem of collection-based image classification. Examples of collection-based image data include 3D shapes and fingerprints [123, 124], RGB-D objects [125], and citizen-science-enabled real-world dataset of plants [126], car models [127], and insects [128]. In these datasets, each item is associated with multiple images that depict different views or perspectives of the same object, *e.g.* front view and side view in 3D shapes; flower and stem of one plant species. Visual representations are extracted from the images using pre-trained CNN models and combined for classification via operations such as concatenation [124, 129–131], maximum [127], weighted sum [123], or a Recurrent Neural Network (RNN) layer [132]. Another approach to combine views for classification is performing standard classification on each image using pre-trained CNN models and combine the softmax scores via summation [133] or multiplication [134]. Seeland and Mäder [135] conducted a systematic review of collection-based image classification approaches and highlighted the benefits of integrating visual representations rather than post-processing the softmax scores.

2.3.2 Multi-modal fusion

Multimodal systems process inputs from more than one modality such as visual, audio, and text that describe the same concept. Multi-modal analysis has attracted increasing attention in recent years with the growing availability of multi-modal data [136–138]. Survey studies have been conducted in the field of multimodal analysis [139–141], and most of the collected works suggested the superiority of multimodal over unimodal approaches. Specifically for the fusion of visual and textual data, a large amount of research and applications have been developed including image annotation [142], image captioning [143], visual question answering [144, 145], as well as grounding textual representations with the visual world [146, 147]. A major body of multi-modal fusion works can be categorized as operation-based fusion, where operations such as concatenation [148–151], averaging [152], weighted summation with scalar weights [153] were used to combine uni-modal represen-

tations for multi-modal tasks. Operation-based fusion introduces few or no additional parameters and is applicable to almost any backbone uni-modal feature extractors. Although these methods are simple and straightforward for implementation, the correlations between different modalities were not considered. Model-based fusion has also been studied where model architecture was designed for fusion. The Multiple Kernel Learning (MKL) extends from kernel Support Vector Machine (SVM) and adopts different kernel functions for different modalities to better fuse heterogeneous inputs [154]. Attention mechanisms built upon neural networks have also been developed for multi-modal fusion tasks such as Visual Question Answering (VQA) or image-text matching that requires attending to the objects in images and words in texts strategically in order to obtain the target output. Attention mechanisms typically consist of operations such as multiplication, dot product, and linear neural network layers that introduce additional trainable parameters manage the multi-modal features for fusion [155–158]. While originally developed for textual data, the self-attention modules have also been adopted for multi-modal fusion, in both single-stream (one module to learn intra-modality representations [159]) and two-stream (modality-specific attention modules followed by a cross-modality module) [160, 161]) manners. Model-based fusion usually requires a data point having one input from each modality, which limits the applicability of existing models for fusing visual and textual data from bridge inspection reports that each has multiple images and sentences with no explicit alignment. The existing attention and self-attention mechanisms focus on modeling the complex relationship among objects from images and words from sentences, while this study focuses on the aggregating of images and sentence representations to support the task of the automatic condition rating.

2.3.3 Domain applications

In the domain of bridge infrastructure management, vision-based condition assessment models have been extensively studied in recent years [44–48]. These vision-based models are focused on the recognition of local deficiencies such as crack [49, 50], delamination [51], and spalling [52] from various surface materials such as steel [53], and asphalt and concrete [54–56]. The condition assessment tasks were formulated as classification at the image/patch level, detection at the object level, or segmentation at the pixel level. The pixel-level defect segmentation models improve the resolution of defect recognition to each pixel within the images, but the pixel scale in real-world coordinates is still unknown unless a scale reference is provided in the images. While the vision-based defect

models recognize local deficiencies at different levels, these outcomes have not yet been aggregated to provide an overall assessment of bridges and support maintenance decision-making. The textual data from bridge inspection reports have also been used to support automated condition assessment. Local condition information (*e.g.* defects, their locations and severity) and their dependency relations have been extracted from bridge inspection reports [39, 40, 162]. The extracted mentions of the same bridge components or defects with various languages have been normalized, and the extracted measures of the same defect type have been fused to support bridge deterioration prediction [163]. Deep learning-based automated condition rating and quality control models have been constructed that summarizes and maps information from descriptive sentences from bridge inspection reports to the overall condition rating [164]. Regarding the fusion of multi-modal data for bridge condition assessment and management, Sun et al. [165] proposed to align the 3D laser scanning data and the textual data from bridge inspection reports to assist bridge inspection and the tracking of bridge defect changes. While an increasing volume of multi-modal data is available from both robotic inspections [30–33] and historical documentation to support bridge infrastructure condition assessment, the fusion of multiple modalities has not been extensively studied in this research community.

Chapter 3

Mapping Textual Descriptions to Condition Ratings to Assist Bridge Inspection and Condition Assessment Using Hierarchical Attention

Li, T., M. Alipour, and D.K. Harris, Mapping Textual Descriptions to Condition Ratings to Assist Bridge Inspection and Condition Assessment Using Hierarchical Attention. *Automation in Construction* (in review), 2020.

3.1 Abstract

Effective upkeep of aging infrastructure with limited resources requires intelligent management systems supported by accurate condition evaluations. Current bridge management strategies rely on experience-driven manually-assigned condition ratings. To improve the accuracy and consistency of these ratings, this study identifies narrative descriptions from bridge inspection reports as an untapped data source and proposes a data-driven framework to map natural language descriptions to quantitative ratings. A hierarchical architecture employing recurrent neural network encoders with an attention mechanism was developed using a collection of reports from the Virginia Department of Transportation, which outperformed a variety of baseline systems. Visualization of the resulting attention patterns was shown to provide interpretable insights which highlight potentially-overlooked

indicators. Application of the system via two use cases further showed the potential of the system as a supportive tool for automated rating recommendation as well as real-time quality control, which can proactively increase the consistency of condition rating practices.

3.2 Introduction

The U.S. bridge infrastructure system is in critical need of maintenance, rehabilitation, and repair (MR&R) operations due to the challenges characterized by aging and deterioration. Based on the latest National Bridge Inventory (NBI) database, nearly forty percent of the U.S. bridge population have approached or passed their design life of 50 years, and over 10% have weight limit posting due to deterioration that restricts heavy traffic loads [20]. One out of every three U.S. bridges has been identified as in need of structural repair, rehabilitation, or replacement and the total cost of these improvements has been estimated at nearly \$164 billion [166]. Effective upkeep of such an aging and deteriorating bridge network with limited funding and resources calls for smart bridge management systems that can accurately capture the deterioration condition of bridges and effectively prioritize the maintenance needs based on the conditions.

Assigning condition ratings properly and consistently is a challenging task mainly for the following two reasons. First, the rating guidelines need to be properly interpreted by individual inspectors in order to assign a rating. Second, the component ratings are an overall representation of local deficiencies, which requires bridge inspection expertise to aggregate local conditions to a holistic component-level rating. Regarding the individual interpretation, the transition between condition ratings does not involve a crisp boundary and depends on inspectors' knowledge and experience. Inspectors refer to national and state-developed supplemental component condition rating guidelines [24, 25] for guidelines and specific procedures in assigning the ratings. With respect to the overall representation of local deficiencies, the FHWA Coding Guideline states that the component condition ratings should reflect the overall conditions of the component rather than localized conditions [1]. This makes the assignment of an overall condition rating based on individual member conditions challenging. The inspectors rely on their bridge engineering expertise to determine the impact of local deficiencies on the overall component condition when assigning component condition ratings. Assigning condition ratings consistently and accurately is a challenging task and innovation is required in the development of intelligent methods of rating estimation and quality control.

Since assigning proper condition ratings relies heavily on the inspectors' bridge engineering ex-

pertise and experience, it requires significant time, budget, and human efforts for inspector training and additional caution throughout the QC/QA procedures. While the efforts and investments by the DOTs have established sound inspection practices and procedures, because of the experience-guided nature of the overall condition rating process, it is difficult to evaluate the reliability and consistency of ratings derived without considering the root of their derivation (*i.e.* the descriptions used to inform the ratings) in aggregate. Research efforts have also been made to perform anomaly detection on the condition rating data, in order to reveal potential spatial or temporal inconsistencies [15, 167, 168] or incompatibilities in attribute properties [169]. However, the majority of the existing research approaches are based on manipulating the inspector-driven condition ratings, rather than directly examining the condition of the infrastructure that derived the ratings.

Although challenges exist in both condition rating and quality control to ensure consistent and reliable bridge condition assessment, the carefully-maintained, massive historical bridge inspection reports contain an untapped source of raw data (*i.e.* narrative natural language descriptions) that can be exploited to support data-driven approaches that map the narrative descriptions of bridge condition to condition ratings. The narrative descriptions from bridge inspection reports contain valuable information for supporting bridge condition assessment in their massive volume, rich details, and the embedded collective inspector domain knowledge and expertise. Bridge inspection reports have been maintained by many state Departments of Transportation (DOTs) for decades, documenting first-hand local condition details in the form of narrative descriptions of bridges as well as condition ratings assigned by experienced inspectors. These inspection reports and the associated ratings collected over the years represent a knowledge base that aggregates individual inspectors' expertise, and can shed light on how narrative descriptions of local deficiencies map to condition ratings. Recent developments in Natural Language Processing (NLP) techniques enable approaches to process textual data, extract information from raw textual data, as well as to summarize and draw conclusions based on the extracted information. However, the application of NLP techniques to bridge inspection and management has been limited [39, 42, 170, 171], creating a demand and an opportunity to utilize these techniques to automatically interpret the untapped narrative descriptions embedded in bridge inspection reports and translate these descriptions into consistent and reliable ratings.

To bridge the gap, this study proposes a data-driven framework that establishes a mapping between the inspection report texts and the condition ratings using deep neural networks. The frame-

work is envisioned to support two application scenarios: 1) mapping the narrative descriptions of bridge condition to condition ratings, and 2) detecting possible inconsistent ratings in a quality control pipeline. In this framework, the model learns from the collective expertise and knowledge embedded in massive historical reports, and hence promotes a unified and consistent approach for assigning condition ratings. Central to this hypothesis is the assumption that the raw narrative descriptions are more objective and subject to less variability from an inspector's experience-driven judgment as opposed to the ratings derived from these descriptions. Unlike the condition ratings which require substantial training and experience to be properly and consistently assigned, describing observations of bridge conditions as seen during an inspection is expected to involve less subjectivity and variability. During field inspection and evaluation, inspectors can utilize the model-recommended rating to help reach a more objective and data-driven decision. Ratings provided by the inspectors can also be run through the proposed quality control pipeline and suspicious candidate ratings will be screened and filtered. This application provides a complementary tool for quality assurance and control (QA/QC) of condition assessment process and hence can proactively increase the consistency of condition rating practices. As an additional insight, the proposed framework can also identify the sentences, phrases and keywords that were paid stronger attention by the model during the NLP process. These indicators can help inspectors understand the logic of a model-generated rating and act as pointers and reminders that may be of value in the rating process. A potential future application of the proposed system is envisioned to be in the development of smart voice-controlled inspection assistants, which can help inspectors perform hands-free documentation of their observation, perform voice queries and information look-up through the database during an inspection. Such an assistant can facilitate the collection of inspection data, assign or recommend condition ratings, and improve the safety of inspection operations which are often performed at heights or in dangerous environments.

Quality Assurance and Control of Civil Infrastructure Data The accuracy and uniformity of the recorded infrastructure conditions are vital for supporting appropriate infrastructure management decisions for preservation, retrofit, replacement, and public safety. Studies show that both systematic and random errors in condition data can highly distort management system outputs such as projected budgetary needs and planned maintenance activities [172]. Besides the regulatory QC/QA procedures such as the required qualifications of the inspection team, periodic refresher training, report reviews, and field review processes [11, 12], extensive research efforts have been

made to assess and improve the quality of the collected infrastructure condition data. A major line of such research involves identifying inconsistency or abnormality in the collected data and improving the quality of the data by addressing the inconsistent or abnormal data points. Buchheit et al. [13] adopted different data quality assessment methods including statistical methods, clustering, pattern-based detection in a voting scheme to improve the quality of civil infrastructure condition data. Outlier and error detection methods were also developed for roadway and pavement infrastructure condition data using spatial and temporal attributes of the condition data [167, 168]. Specific to the National Bridge Inventory, Din and Tang [15] developed logical tests to reveal the temporal and spatial anomalies in the NBI data, where temporal anomalies were revealed by comparing data from different years to check for logical errors, and spatial anomalies were identified by geospatial mapping to identify spatial conflicts. Chen et al. [169] developed a Web-based tool that identifies anomalies using a set of pre-defined rules and provides feedback about the detected anomalies. The tool was then applied to the Pennsylvania Department of Transportation (PennDOT) bridge management dataset, which has similar data items as those in the NBI data, and was able to detect incompatibilities with the requirements or codes. Management practices and protocols have also been proposed to improve the quality of infrastructure condition data. Migliaccio et al. [14] proposed a condition data collection procedure that adopts Agreement Between Evaluators (ABE) and Consistency Over Time (COV) assessments to reduce the variance among individual condition ratings and to prevent inconsistent conditions before and after maintenance actions.

The efforts reviewed in the previous sections, both regulatory and in research, have contributed to improvements in quality assurance and control of civil infrastructure data. However, the existing approaches still have limitations in the following two aspects. First, most of the studies (except [14]) are retrospectively designed for an already-established condition database instead of quality control in real-time as the condition data was collected in an ongoing inspection. Anomaly or data outliers were detected to be excluded from future maintenance planning analysis, without offering the opportunity to refer back to the cause of each specific error, or to improve the condition evaluation practice by learning from the detected errors. Second, the majority of existing approaches took an anomaly detection viewpoint by looking at the assigned condition scores itself (together with related spatial or temporal features), instead of looking directly at the underlying condition of the infrastructure evaluated. This might be partially attributed to the fact that, although the NBI database provides bridge characteristics and condition ratings, it does not keep track of the details

of bridge conditions such as local defects and their severity. Therefore, the NBI database provides limited support for developing an automated QC/QA program. In this regard, the rich and detailed information from bridge inspection reports introduces an opportunity to improve the quality of condition evaluation. Another advantage of using the inspection reports is that these reports are already required in the current bridge inspection practice and have already been collected for years for other purposes. The massive amount of ‘report narratives + condition ratings’ pairs represent a knowledge base that is embedded with the expertise of different inspectors regarding how a condition rating is assigned based on their past training and experience. Nevertheless, this valuable information has remained untapped as it relates to the application of automated condition evaluation due to the challenges in extracting condition-related information from the natural language narrative form of the reports that were composed by individual inspectors with various narration preferences. This warrants further studies on how to exploit the untapped narrative description from bridge inspection reports to improve the quality of bridge infrastructure condition data and ultimately infrastructure management decisions.

3.3 Proposed Research

This study proposes to exploit bridge inspection reports for automated condition rating and quality control considering the following advantages: 1) the inspection reports are already required in the current bridge inspection practice and have already been collected for years for other purposes; 2) narrative descriptions in these reports documents the details of bridge local deficiencies; 3) the massive amount of ‘report narratives + condition ratings’ pairs represent a knowledge base that is embedded with the expertise of different inspectors regarding how a condition rating is assigned based on their past training and experience. Given the rich condition details and the domain expertise contained in the inspection reports, this study proposes a data-driven framework capable of learning the mapping from the narrative descriptions within historical inspection reports to bridge condition ratings. A large corpus of bridge inspection reports were collected from the Virginia Department of Transportation (VDOT) to develop the proposed framework, which includes over 8,000 reports from 9 districts in the Commonwealth of Virginia. Other state DOTs also maintain similar report records that can offer potential extension of this work to a national scale. The proposed framework leverages the individual narratives from the entire population of bridges within the VDOT inspection inventory and their associated condition ratings to formulate a model capa-

ble of translating unused descriptive narratives into numerical condition ratings. The fundamental premise of this work is that although assigning a condition rating properly and consistently is a challenging task that depends on the inspector’s knowledge and experience, generating descriptions of the conditions as seen during an inspection is much easier and less sensitive to subjectivity. Furthermore, patterns between the descriptions and ratings extracted from the collective decisions by many inspectors can provide valuable insights into consistent condition rating.

The review of text classification applications in the domain of civil infrastructure management showed that the valuable information from bridge inspection reports has not been used to support automated condition evaluation. One possible challenge for such application lies in extracting condition-related information from the natural language narrative form of the reports that were composed by individual inspectors with various narration preferences. This warrants further studies on how to exploit the untapped narrative description from bridge inspection reports to improve the quality of bridge infrastructure condition data and ultimately infrastructure management decisions. While the domain text classification applications contributed to the advancement of automation in construction and infrastructure management, the gaps in the existing applied models can be summarized into the following three aspects.

- First, the majority of the reviewed studies used word-frequency-based features such as TF-IDF, TFRF, and $TF_{max}RF$ to support text classification models, which capture limited semantic information from the textual data. Given the linguistic theory that semantic information lies in the co-occurrence of words [92], the frequency-based features capture a global word co-occurrence pattern in the sentence or document to be classified, and hence contain a certain level of semantic information. However, transforming a sentence or document into word frequencies results in the loss of the local contextual information of each word. Few studies, except [112, 113], used dense word embeddings that are encoded with semantic and contextual information.
- Second, the use of context-aware deep-learning-based methods is scarce in the field of infrastructure management and maintenance, even though the complexity of the textual data from infrastructure inspection reports demands such models to capture the various usages of words and extract correct information. This need is highlighted by the highly heterogeneous nature of the inspection reports. Infrastructure inspection reports are created by various professional inspectors with no standard requirements on the type of language and wording to use in the

narratives, and contain a mixture of technical and non-technical words. Being able to model the semantics and contexts of the words is essential for supporting the successful analysis of complicated inspection reports.

- Third, existing classification applications focus on categorizing textual data from dissimilar types, *e.g.* Caldas et al. [103] categorized construction project documents into types such as scheduling, HVAC, and fire protection. However, the existing applications did not focus on the problem of assigning a rating score to texts of the same type. Textual data with contents of different types tend to have different sets of words commonly used in each type, which can be leveraged in training a classification model. However, narrative descriptions from bridge inspection reports contain a somewhat similar set of words describing bridges and their conditions. Assigning a rating score requires a model to have a better understanding in the semantics of the textual inputs in order to support the inference of a rating score. Whether a classification model can differentiate between descriptions of different severity of bridge deficiencies to support automatic rating is still unknown and warrants investigation.

Motivated by these gaps, this study proposes a deep neural network building upon dense word embedding features to fully exploit the semantics and context of the heterogeneous narrative descriptions from bridge inspection reports. As illustrated in Figure 3.1, the narrative descriptions from one report can be viewed as a document that contains L sentences ($sentence_i, i \in [1, L]$), where $sentence_i$ contains T_i words, denoted by $word_{it}, t \in [1, T_i]$. Each sentence describes an item of the inspector's findings during a field inspection, as illustrated in Figure 1.2 with the corresponding local conditions. Motivated by the hierarchical structure of document-sentences-words, the proposed framework develops a hierarchical attention network that progressively summarizes from the word-level to the sentence-level and ultimately to a document-level representation, which can then be used for mapping to the condition ratings. The hierarchical attention mechanism offers interpretability as to which sentences and words were emphasized while assigning a condition rating. Taking the narrative description from one bridge inspection report as inputs, the hierarchical attention network outputs a series of softmax probability scores, one for each condition rating category. Two applications are developed using these scores: 1) a condition rating tool that recommends the rating with the maximum score and 2) a quality control tool that takes an inspector-provided rating as an additional input, computes a log-likelihood ratio, and compares the ratio with a data-driven decision threshold Θ to generate the decision of whether to accept or reject the provided rating. The

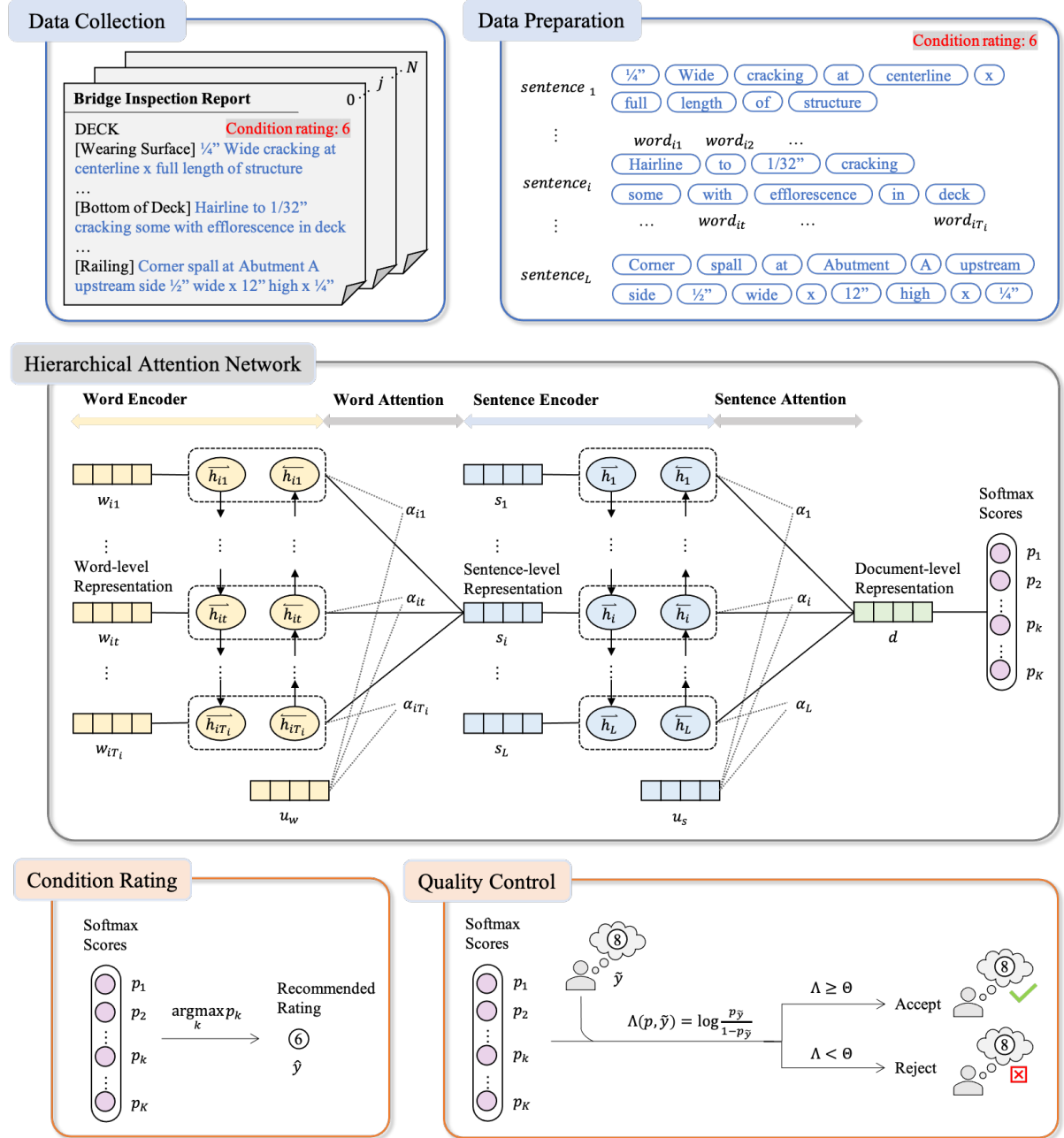


Figure 3.1: The proposed framework for automatic condition rating and quality control.

condition rating tool directly translates the narrative descriptions into recommended condition ratings, while the quality control tool compares inspector-assigned ratings to historical practice learned by a data-driven model to promote consistency in the rating practice. The quality control application links the data-driven model directly to decision-making while keeping the human inspector in the decision loop, which is critical in the field of civil infrastructure management considering the importance of structural safety and the potential impact of infrastructure failure.

3.4 Research Methodology

This section is organized to present the proposed framework as follows: first, the Gated Recurrent Unit (GRU) sequence encoder, which is adopted in the proposed HAN model, is presented; then the architecture of the HAN model is discussed, followed by a description of the two proposed applications: recommending a condition rating and quality control via accepting or rejecting an inspector-provided condition rating.

3.4.1 GRU-based Sequence Encoder

The learning module of the proposed framework utilizes a hierarchical attention network to establish document-level representations progressively from the word-level and sentence-level. Before being fed through the attention mechanism, the sequences of word-level and sentence-level representations are first encoded with contextual information using a sequence encoder. The Gated Recurrent Unit (GRU) [114] is a special type of Recurrent Neural Network (RNN) that enables the encoding process by introducing recurrent connections between the nodes in the hidden layers to model the dependencies among sequential inputs. The GRU node uses a gating mechanism to control the flow of information among a sequence of inputs. Two types of gates exist in the gating mechanism including the reset gate r_t and the update gate u_t . Denoting the input at time step t as x_t and the output of a GRU node at time step $t - 1$ as h_{t-1} , the reset gate is computed by Equation 3.1

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r) \quad (3.1)$$

where σ is the element-wise logistic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and the W_{rh} , W_{rx} , and b_r are the corresponding weight matrices and bias for the reset gate, respectively. Similarly, the update gate is computed by Equation 3.2

$$u_t = \sigma(W_{uh}h_{t-1} + W_{ux}x_t + b_u) \quad (3.2)$$

where W_{uh} , W_{ux} , and b_u denote the weight matrices and bias for the update gate. The formulation of the GRU gating mechanism is presented in Equation 3.3,

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot \tanh(W_{hh}(r_t \odot h_{t-1}) + W_{hx}x_t + b_h) \quad (3.3)$$

where h_t is the output of a GRU node at time step t , W_{hh} , W_{hx} and b_h denote the corresponding weight matrices and bias, and \odot represents element-wise multiplication. Figure 3.2 presents the gating mechanism inside a GRU node. The reset gate is formulated to control how much information from the previous hidden state h_{t-1} contributes to the current h_t . The update gate u_t controls how much information from the previous output is kept and how much is added after the reset gate.

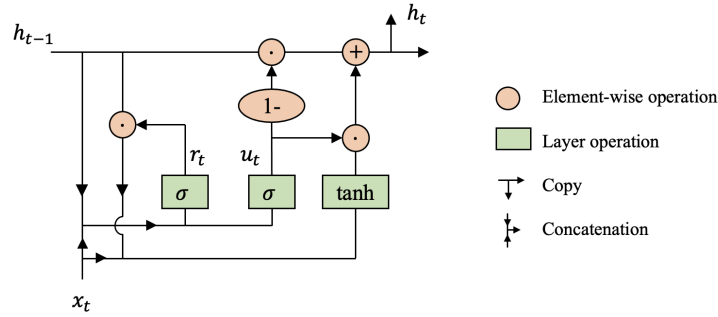


Figure 3.2: The gating mechanism inside a GRU node.

3.4.2 Hierarchical Attention

Since the narrative descriptions from each bridge inspection report can be treated as a document that contains sentences composed of words, this study develops a hierarchical attention network that progressively summarizes information from word-level vector representation to the sentence-level and ultimately to the document-level representation, which can then be used to form the mapping from the narrative descriptions to the condition ratings. The hierarchical attention mechanism also offers interpretability as to which sentences and words were emphasized while assigning a condition rating. Each component of the hierarchical attention architecture is introduced in the following subsections including the word-level encoder, word-level attention, sentence-level encoder, and sentence-level attention.

Word Encoder For the narrative descriptions from each inspection report, assume it contains L sentences $s_i, i \in [1, L]$ and each sentence contains T_i words. The words from sentence s_i , denoted

by $w_{it}, t \in [1, T_i]$, are first embedded into a vector representation using an embedding matrix W_e , where each row is a vector for a word from the textual data. The embedding matrix can either be randomly initialized, or can use embeddings previously developed using large corpora [70, 71]. Each word w_{it} is then represented by a vector, which is the corresponding row vector taken from the embedding matrix W_e . Given a sequence of word representations $w_{it}, t \in [1, T_i]$, a bidirectional GRU [114] encoding layer is used to exploit the dependencies within the sequence and encode contextual information from both directions to a new sequence of summarized representations. The bidirectional GRU contains a forward pass \overrightarrow{GRU} that reads the sentence s_i from w_{i1} to w_{iT_i} , and a backward pass \overleftarrow{GRU} that reads the sentence s_i from w_{iT_i} to w_{i1} (as presented in Equation 3.4, 3.5). The bidirectionality ensures that contextual information from both before and after each word is encoded.

$$\overrightarrow{h_{it}} = \overrightarrow{GRU}(w_{it}), t \in [1, T_i] \quad (3.4)$$

$$\overleftarrow{h_{it}} = \overleftarrow{GRU}(w_{it}), t \in [T_i, 1] \quad (3.5)$$

The encoded vector, denoted by h_{it} for word w_{it} is obtained by concatenating the outputs from both forward and backward passes $h_{it} = [\overrightarrow{h_{it}}; \overleftarrow{h_{it}}]$.

Word Attention Since the words from a sentence are not equally important in conveying the meaning of the sentence, different levels of attention should be paid to each word when summarizing and aggregating the word representations to form a sentence representation. To that end, word attention introduces a mechanism formulated as

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (3.6)$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \quad (3.7)$$

where u_{it} denotes a hidden representation of h_{it} obtained by feeding h_{it} through a simple fully connected layer (W_w and b_w denote the corresponding weight and bias parameters, respectively.) The importance of a word w_{it} is measured by the similarity between u_{it} and a word-level context vector u_w , and is then used to compute a normalized word attention weight α_{it} using a softmax function as presented in Equation 3.7. In this attention formulation, u_w is designed as the representation of an overall informative word, where the attention weight for each word is computed based on the word's similarity with u_w . The context vector u_w is a randomly initialized vector and is learned during the

training process together with other parameters incorporated in the attention mechanism. The sentence vector s_i is computed by a weighted sum of the encoded word representations as presented in Equation 3.8.

$$s_i = \sum_{t=1}^{T_i} \alpha_{it} h_{it} \quad (3.8)$$

Sentence Encoder Given a sequence of sentence vectors $s_i, i \in [1, L]$ resulting from the word attention step, the sentence encoder encodes each vector with contextual information from both before and after each sentence using a bidirectional GRU in a similar fashion as the word encoder:

$$\vec{h}_i = \overrightarrow{GRU}(s_i), i \in [1, L] \quad (3.9)$$

$$\overleftarrow{h}_i = \overleftarrow{GRU}(s_i), i \in [L, 1] \quad (3.10)$$

The encoded representation h_i is then obtained by concatenating \vec{h}_i and \overleftarrow{h}_i : $h_i = [\vec{h}_i; \overleftarrow{h}_i]$.

Sentence Attention The sentence attention further summarizes and aggregates the encoded sentence representations $h_i, i \in [1, L]$ into a document level representation by assigning different attention weight to each sentences, as presented in Equation 3.11 and 3.12.

$$u_i = \tanh(W_s h_i + b_s) \quad (3.11)$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \quad (3.12)$$

where u_i denotes a hidden representation of h_i obtained by feeding h_i through a fully connected layer (W_s and b_s denotes the corresponding weight and bias parameters, respectively.) The sentence attention weight α_i computes normalized sentence importance, which is measured by the similarity between u_i and a sentence-level context vector u_s . The context vector u_s is also a randomly initialized vector that is learned during the training process. At last, the document vector d is computed by a weighted sum of the encoded sentence representations as presented in Equation 3.13.

$$d = \sum_{i=1}^L \alpha_i h_i \quad (3.13)$$

3.4.3 Condition Rating

The document-level vector d provides a high-level representation of extracted information from the bridge inspection reports, and is used as features for mapping to the condition ratings, as presented

in Equation 3.14 and 3.15.

$$q = W_d d + b_d \quad (3.14)$$

$$p_k = \frac{\exp(q_k)}{\sum_{k'=1}^K \exp(q_{k'})}, \forall 1 \leq k \leq K \quad (3.15)$$

where the document vector v is fed through a fully connected layer (W_d and b_d denote the corresponding weight and bias parameters), $q \in \mathbb{R}^K$ where q_k represents the k_{th} element in q , and K is the total number of condition rating categories. The vector $p \in \mathbb{R}^K$ consists of p_k , which computes the probability of the condition being rated as k . Given the target condition rating $y \in \{1, 2, \dots, K\}$, the cross entropy loss used for model training is presented in Equation 3.16.

$$L(p, y) = -\log(p_y) \quad (3.16)$$

To assign a condition rating, the model computes the predicted rating \hat{y} as presented in Equation 3.17.

$$\hat{y} = \arg \max_k p_k \quad (3.17)$$

Considering the ordinal nature of the condition ratings (Table 1.1), mis-predicting a rating on the order of a single condition state (*e.g.* predicting of 4 instead of 5) should cause a different level of concerns as compared to mis-predicting across multiple condition states (*e.g.* predicting of 4 instead of 9). Therefore, for evaluating the condition rating performance, metrics that are commonly used for evaluating ordinal outputs were used, including Accuracy (ACC0 and ACC1), Mean Squared Error (MSE), and Mean Absolute Error (MAE) [173], as computed in Equation 3.18, 3.19, 3.20, and 3.21, respectively.

$$ACC0 = \frac{1}{N} \sum_{j=1}^N \mathbb{1}(\hat{y}_j, y_j) \quad (3.18)$$

$$ACC1 = \frac{1}{N} \sum_{j=1}^N [\mathbb{1}(\hat{y}_j, y_j) + \mathbb{1}(\hat{y}_j, y_j + 1) + \mathbb{1}(\hat{y}_j, y_j - 1)] \quad (3.19)$$

$$MSE = \frac{1}{N} \sum_{j=1}^N (\hat{y}_j - y_j)^2 \quad (3.20)$$

$$MAE = \frac{1}{N} \sum_{j=1}^N |\hat{y}_j - y_j| \quad (3.21)$$

where $\mathbb{1}(\cdot, \cdot)$ is the indicator function, y_d and \hat{y}_d denote the target condition rating and the predicted rating of document d respectively, and N denotes the total number of data items to be evaluated. In

addition to the ACC0 that evaluates the exact rating accuracy, ACC1 was used to compute accuracy within 1 level of the true condition ratings, and the MSE and MAE evaluate an averaged error of the predicted ratings.

3.4.4 Quality Control

This section proposes a quality control process that evaluates the inspector-provided rating based on its narrative descriptions and generates a decision of whether to accept or reject the provided rating. With an inspector-provided rating \tilde{y} and a model-predicted rating \hat{y} as computed in Equation 3.17, the direct way for quality control is to accept the candidate rating when $\hat{y} = \tilde{y}$ and to reject it when $\hat{y} \neq \tilde{y}$. However, this direct way does not take into consideration the probability scores $p_k, \forall 1 \leq k \leq K$ that led to the model prediction of \hat{y} .

Table 3.1 presents two example cases of probability scores. In both cases, the model predicted that the rating \hat{y} should be condition 6 (maximum class probability in each case). Suppose in both cases the inspector assigned a condition rating of 5, the direct way of quality control would reject the provided rating in both cases but with very different certainty. In case 1, p_6 is only slightly larger than p_5 which indicates that the model is less certain of predicting condition 6 compared with case 2, where p_6 is much higher than p_5 . To ensure that the acceptance or rejection decision of each provided condition rating \tilde{y} is generated with the same level of certainty, a Likelihood Ratio Λ is computed as presented in Equation 3.22.

$$\Lambda(p, \tilde{y}) = \log \left(\frac{p_{\tilde{y}}}{1 - p_{\tilde{y}}} \right) \quad (3.22)$$

Table 3.1: Two example cases of model-generated probability scores with the inspector-assigned rating of 5.

Condition Rating		4	5	6	7	8	9
p_k	Case 1	0.10	0.34	0.36	0.10	0.10	0.00
	Case 2	0.00	0.01	0.90	0.09	0.00	0.00

This likelihood ratio estimates the relative confidence of the model in the inspector-provided rating compared with other ratings. Assuming the threshold is denoted as Θ , the model accepts the provided rating \tilde{y} when $\Lambda \geq \Theta$ and rejects the provided rating \tilde{y} when $\Lambda \leq \Theta$. In other words, the decision is not solely based on the class with the highest probability, but instead based on

achieving a desired level of relative confidence. For the two example cases in Table 3.1, with a decision threshold $\Theta = -2$, $\Lambda_{case1} = \log\left(\frac{0.34}{1-0.34}\right) > \Theta$ and hence the inspector-assigned rating can be accepted; however, $\Lambda_{case2} = \log\left(\frac{0.01}{1-0.01}\right) < \Theta$ and hence the inspector-assigned rating is still rejected. The threshold Θ is a data-driven hyper-parameter that can be tuned before application. For a bridge management agency intending to adopt this tool for quality control of their ratings, a small portion of its bridges can be used as a validation set to tune the hyper-parameter Θ . For this validation set, ground-truth condition ratings should be provided by experienced inspectors, and the candidate ratings assigned by in-training inspectors together with the underlying narrative descriptions are fed through the system to obtain a decision of "accepted/rejected". The optimal Θ value can be tuned using this validation set by finding the threshold at which the performance is maximized in alignment with the ground-truth ratings from experienced inspectors. For evaluating the quality control performance, metrics including Precision (PRE), Recall (REC), and $F1$ score can be used, as presented in Equation 3.23, 3.24, and 3.25.

$$PRE = \frac{T_P}{T_P + F_P} \quad (3.23)$$

$$REC = \frac{T_P}{T_P + F_N} \quad (3.24)$$

$$F1 = \frac{2 \times PRE \times REC}{PRE + REC} \quad (3.25)$$

where T_P denotes the number of provided ratings correctly accepted by the model, F_P denotes the number of provided ratings falsely accepted by the model, and F_N denotes the number of provided ratings falsely rejected by the model. Precision computes of all the provided ratings accepted by the models, how many were actually correct; and recall computes of all the correct provided ratings, how many were accepted by the model. The $F1$ score is the harmonic mean of precision and recall, which evaluates the overall performance of accepting or rejecting the provided ratings. Once the hyper-parameter Θ is tuned based on these metrics evaluated on the validation set, incoming condition ratings together with the underlying narrative descriptions can be fed into the system and a recommendation is made by the system as to whether or not the assigned rating is in alignment with the data.

3.5 Data Collection and Preparation

Bridge inspection reports were collected from the Virginia Department of Transportation (VDOT) database. The reports on the first layer of the VDOT database were downloaded in bulk in June 2018, which mostly contain the most recent inspection reports for all bridges across the state of Virginia. While historical versions of the inspection reports were also available through the database in deeper layers, their collection involves a manual process of individual downloading rendering the collection process prohibitively cumbersome.

Each report was organized by sections detailing different bridge components such as "Deck", "Superstructure", and "Substructure", and each bridge component was associated with a condition rating of 0-9 assigned by the inspection personnel. The general condition of bridges can also be categorized as good, fair, and poor, which denotes condition ratings greater than 6, equal to 5 or 6, and less than 5, respectively [174]. Although the inspection reports typically include the assigned condition ratings on the first page, it is hard to directly extract the condition ratings from the collected reports since a large portion of the collected reports use a photo-format cover page to include the condition rating. Therefore, of all the collected reports, this study used the reports of NBI bridges (bridges more than 20 feet long used for vehicular traffic [1]), whose condition ratings can be obtained from the NBI database [1]. The federal bridge ID and the date of inspection were extracted from the inspection reports, usually in the first sentence of the report text, using regular-expression matching. The extracted federal bridge ID and date of inspection were then used to match the inspection report with the associated condition rating from the NBI data.

Since each inspection report contains a large table of all its contents, the narrative descriptions were extracted from the inspection report files using the python-docx package [175], where irrelevant details such as table frames, document headers and footers, images and their captions were excluded. The python-docx package supports reading the tables row by row from the Word document of inspection reports and extracting the narrative descriptions from each cell in the table. This allows keeping the extracted narrative descriptions in the same organization as originally in the tables of the report files. The section of bridge deck condition description and a corresponding deck condition rating were used to construct the rating model in this study. Similar models can also be trained for other bridge components (such as superstructure or substructure) to learn the mapping from narrative descriptions to their condition ratings using the proposed method. Considering that

only 1.64% of the bridge decks in Virginia were in poor condition as of the year 2018, reports of 200 bridges that have historically been rated as condition 4 were manually downloaded from the VDOT database and added to the collected data.

For the 10,521 NBI bridges in Virginia [176], the above process resulted in the collection of narrative descriptions of 7,766 unique bridges and a total of 8,028 bridge inspection reports (246 of these bridges had more than one version of reports that were historical reports containing different text and rating scores). The collection process also resulted in 2,755 bridges not included in the collection of this study for reasons including reports in a PDF format; not uploaded to the VDOT database, not listing the date of inspection at the beginning of the report; or not organized by bridge components of "Deck", "Superstructure", "Substructure", *etc.*. Figure 3.3 (top) illustrates the distribution of the collected bridges from nine VDOT regional offices. Each district office adopts several different inspection teams and personnel, and therefore the collected reports incorporate an aggregated knowledge base of inspectors' expertise on how bridge conditions should be mapped to the rating scores. Figure 3.3 (bottom) presents the characteristics of the collected bridges including material, design, and structural deficiency. It can be seen from this figure that roughly half of the collected bridges were steel bridges, while the other half were concrete or pre-cast concrete bridges. Over 60% of the collected bridges were multi-girder bridges, and the rest of the bridges were of

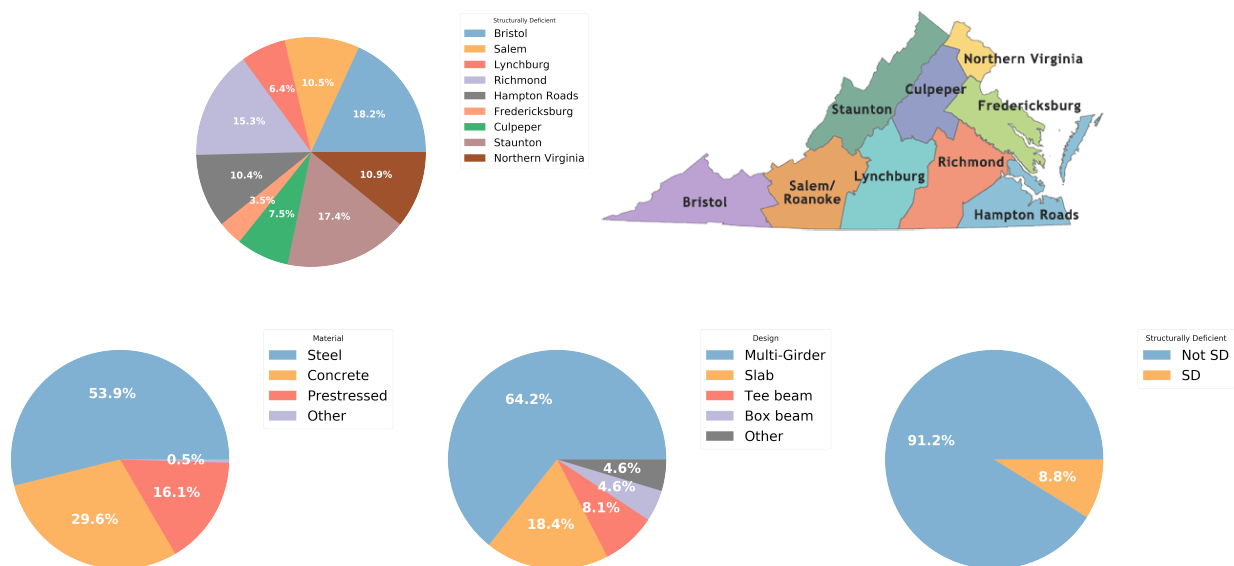


Figure 3.3: Top: Distribution of bridges from nine VDOT regional offices [3]. Bottom: Characteristics of the collected reports including material, design, and structural deficiency (SD: Structurally Deficient).

design types including slab, T-beam, or box-beam bridges. Less than 10% of the bridges were designated as structurally deficient, which is a classification given to bridges with condition ratings of 4 or less for any major bridge components including the deck, superstructure, or substructure [174].

To obtain a clean inspection report corpus, the raw description texts were tokenized into individual words (tokens) using the Natural Language Toolkit (NLTK) [177] tokenizer, which splits the words based on whitespace, standard contractions, and punctuation. Raw tokens were reduced to their lower-case form, and special characters were removed. Engineering conventions of measurements (*e.g.* 1-1/2 inch) were reduced to decimal numbers and floored down to the nearest integer. The cleaned tokens were then joined by a white-space and grouped into sentences using the NLTK sentence tokenizer [177], which was pre-trained using a large corpus to be able to identify the boundaries of the sentences. In the end, each document of bridge deck descriptions was organized in a hierarchical structure that contains sentences of tokens.

3.6 Results and Discussions

Among all the text-rating pairs of bridge deck descriptions, 90% were randomly selected as the training set, while the rest were used for testing. Table 3.2 presents the statistics of the splits. Two principles were followed when creating the training-testing splits for holdout evaluation. First, the training-testing split followed a stratified random sampling regime resulting in a uniform ratio of training and testing samples from each condition category. Second, to achieve full independence between the training and testing sets and ensure realistic evaluation, reports of the same bridge were assigned to either training or testing set. This process resulted in 6,988 bridges in the training set and 788 different bridges in a mutually exclusive testing set. The training data was used for the development of the proposed model, while the testing data was held unseen during the training

Table 3.2: Statistics of the training and testing sets.

Dataset	# Bridges	# 4	# 5	# 6	# 7	# 8	# 9	Average # Sentences	Average # Words per Sentence
Training	6,988	323	1,011	2,156	2,950	667	118	18.4	14.1
Testing	778	36	112	240	328	74	13	18.1	13.7

process to ensure objective evaluation of the final model performance. It should also be noted that bridges with a condition rating of 3 or lower are closed to traffic due to severe structural deficiencies and safety concerns; hence, they do not represent bridges in realistic operational conditions and are not included in the target condition ratings.

The network was trained and optimized using the mini-batch stochastic gradient descent (SGD) approach, which minimizes the loss as defined in Equation 3.16 by descending the parameters along their gradients. The hyper-parameters of training were selected by performing a 5-fold cross validation (CV) grid-search to optimize performance. As a result, a batch size of 64 with a learning rate of 0.05 were selected for this study. A momentum of 0.9 with $1e-6$ weight decay (L2 regularization) was used, which keeps a portion of the previous parameter update to stabilize the learning process [178]. To alleviate over-fitting, dropout training [179] was applied to the embedding layer that randomly drops the embeddings with a probability of 0.5, which is a well-established regularization technique to prevent complex co-adaptation of weights [180]. The dimensions of the word-level and sentence-level GRU were set to be 50. The CV grid-search for hyper-parameter selection was performed over the parameter space discretization of learning rate [0.001, 0.01, 0.05, 0.1], momentum [0, 0.9], weight decay [$1e-6$, $1e-4$, $1e-2$], dropout [0, 0.3, 0.5], and GRU dimension [50, 100]. The embedding matrix W_e was initialized using the Global Vectors (GloVe) [71], which learn semantic information from the global word co-occurrence matrix and generate embeddings with meaningful linear substructures. Parameters in the embedding matrix W_e were allowed to be fine-tuned during the training process to be further adjusted to the task of mapping narrative descriptions to condition ratings. The network was implemented using the PyTorch [181] package which supports tensor computations with GPU acceleration and neural network optimization with automated differentiation. The model training process was deployed using NVIDIA Tesla P100 GPU nodes provided by the University of Virginia’s High-Performance Computing (HPC) servers.

3.6.1 Condition Rating Performance

Figure 3.4 (a) presents the training and validation curves. The validation ACC0 gradually increased and converged after around 35 epochs, while the training curve is increasing throughout the epochs as expected. Figure 3.4 (b) presents the confusion matrix of the testing predictions. The confusion matrix demonstrated a clear trend of diagonal concentration where the model rarely mistakes by more than one level. This is also evidenced by the fact that the percentage of “missing-by-2”

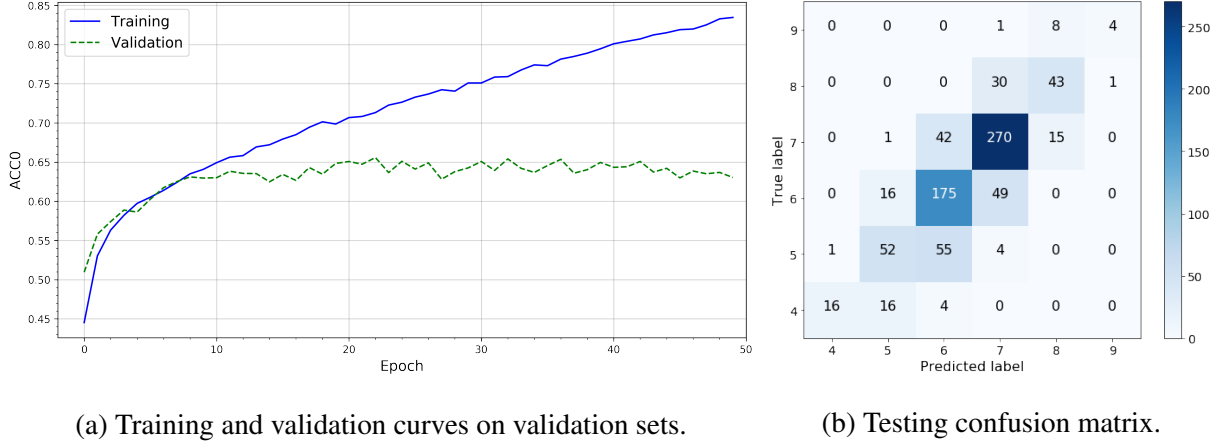


Figure 3.4: Condition rating performance of the proposed hierarchical attention framework.

error (*e.g.*, ground truth condition 9 but predicted as 7) is 1.25%. Similarly, the overall average difference between the ground truth and estimated ratings is 0.339 and 0.315 as measured by the MSE and MAE, respectively (also listed in Table 3.3). This corresponds to ACC0 and ACC1 values of 69.74% and 98.75%, respectively. An examination of some of the error cases will be presented in the next section (together with the corresponding attention maps) to shed light on some of the potential sources of mis-classifications.

It should be noted that while performance evaluation on the testing set (as shown by the confusion matrix in Figure 3.4) demonstrates relatively small MAE and MSE values and high overall ACC0 and ACC1, a closer examination of the confusion matrix indicates increasing difficulties toward the ends of the condition spectrum. A potential explanation for this behavior is the existence of severe imbalance among the size of the classes as seen in Table 3.2, where about 3/4 of the inspection reports belong to the condition categories of 6 and 7, and that categories 9 and 4 have only 1.7% and 4.6% of the data, respectively. Class imbalance is widely known in the machine learning literature to skew the classification performance among the different classes [182–186]. Two major approaches to combat class imbalance include cost-sensitive re-weighting of the mis-classifications [187, 188], and re-sampling the training data (*e.g.*, under-sampling the majority or over-sampling the minority with synthetic samples) [189]. While re-sampling was not practical due to the nature and limited size of the data, the use of re-weighting (median class frequency balancing [190]) did not result in improved accuracy distribution among the classes. This can be attributed to the limited size of the data with only a few hundred data points in the minority classes making it hard to learn the underlying trends even with proper re-balancing. Future work is required to combine

inspection data from multiple states which is expected to allow for effective re-weighting and/or under-sampling of the majority classes.

Table 3.3 presents the performance of the proposed model compared with representative baseline systems from the literature. Based on the review of the existing literature presented before, five most widely used ML baselines that have been consistently used in the literature of civil infrastructure inspection and management [105, 106, 108, 109] were selected, namely Naïve Bayes (NB), Decision Tree (DT), k-Nearest Neighbors (k-NN), Logistic Regression (LR), and Support Vector Machine (SVM). The classifiers in the ML baselines used the term frequency (TF) multiplied by the inverse document frequency (IDF) as features. In this context, TF is the frequency of each term in each inspection report document. The document frequency (DF) of a term is the number of reports that contain the term divided by the total number of documents, and the IDF is the logarithm of the inverse DF. Together, the TF-IDF captures the frequent words that help reveal the gist of each report document while limiting the effect of words that are common to all report documents. In addition to the ML baselines, two deep learning (DL) baselines were also identified from the literature [95, 111], namely LSTM and GRU. The DL baselines used regular recurrent neural networks with LSTM or GRU cells with the same GloVe [71] pre-trained embeddings as the developed system, but did not include the hierarchical architecture or attention mechanisms proposed herein. To ensure a fair comparison, the hyper-parameters of the DL baselines were optimized via grid-search on a grid discretization similar to the proposed method. In this case, the resulting optimal parameter

Table 3.3: Condition rating performance of the proposed framework compared with baseline systems.

Method		ACC0 (%)	ACC1 (%)	MSE	MAE
ML Baselines (TF-IDF)	NB	53.92	91.03	0.786	0.562
	DT	56.66	92.15	0.681	0.514
	k-NN	57.04	95.39	0.580	0.478
	LR	62.52	96.51	0.504	0.415
	SVM	64.88	97.51	0.426	0.376
DL Baselines (GloVe)	LSTM	67.37	98.26	0.379	0.344
	GRU	67.75	97.76	0.390	0.345
Proposed	HAN	69.74	98.75	0.339	0.315

configuration for LSTM was a learning rate of 0.05, momentum of 0.9, and weight decay of $1e-6$, as well as a dropout of 0.6 and cell dimension of 100. The grid search for GRU baseline resulted in the same parameter configuration except for a dropout of 0.3 and weight decay of $1e-4$.

As presented in Table 3.3, the proposed hierarchical attention network (HAN) outperformed the ML and DL baselines across all four metrics. As can be seen in this table, both DL baselines achieved higher performance than the ML baselines across the board, and the developed hierarchical attention network (HAN) further outperformed LSTM and GRU in terms of accuracy metrics. It should also be noted that while the reported values of MAE (and MSE) show that the predictions are on average off only by a fraction of a rating level for all of the methods, the value of MAE for HAN shows relative reductions of 8.7% and 16.3% over the next best DL and ML baselines, respectively (GRU and SVM). In addition to the modeling capabilities of recurrent neural networks, the improvement produced by the proposed model (and the DL baselines) over ML baselines can be attributed to the fact that unlike TFIDF which is based on the frequency features of the text (and thus does not fully incorporate the context and semantics of the documents), the DL solutions leverage embeddings that have been optimized for the specific problem at hand through end-to-end training. It can also be seen that the advantage of the developed HAN model over the GRU baseline is not substantial. This is because both methods use deep learning with the same GRU nodes and embeddings within different architectures. However, a major advantage of the proposed model remains in the additional transparency and interpretability arising from the attention weights which can help inspectors see which words and sentences contributed more to a certain rating recommendation by the model.

Impact of word embeddings Table 3.4 presents the testing performance of the proposed HAN model trained using two pre-trained embeddings: Word2Vec and GloVe. The Word2Vec embeddings were developed using a neural network that encodes a word's dependency with its surrounding words into its embedding, and was trained using the Google News dataset with around 100 billion words [70]. The GloVe embeddings were developed using a global word co-occurrence matrix and were trained using the Wikipedia dataset with 6 billion tokens [71]. As presented in Table 3.4, using GloVe embeddings slightly improved the testing performance in terms of the four metrics compared to using Word2Vec.

Table 3.4: Condition rating performance of the proposed HAN model with different word embeddings.

Embedding	ACC0 (%)	ACC1 (%)	MSE	MAE
Word2Vec	67.37	98.13	0.382	0.345
GloVe	69.74	98.75	0.339	0.315

Impact of dropout To examine the effect of dropout training, the proposed HAN model was trained using different dropout probabilities. Figure 3.5 depicts the training and validation curves for different dropout probabilities. As presented in the figure, using dropout (training with a probability of either 0.3 and 0.5) alleviated over-fitting, where the gap between the training and validation curves is much smaller compared to the case with no dropout. Furthermore, as the dropout probability increases, the validation ACC0 was able to converge to a higher value compared to dropping out with a probability of 0.3, which overall demonstrates the benefit of dropout in terms of improved learning and reduced over-fitting.

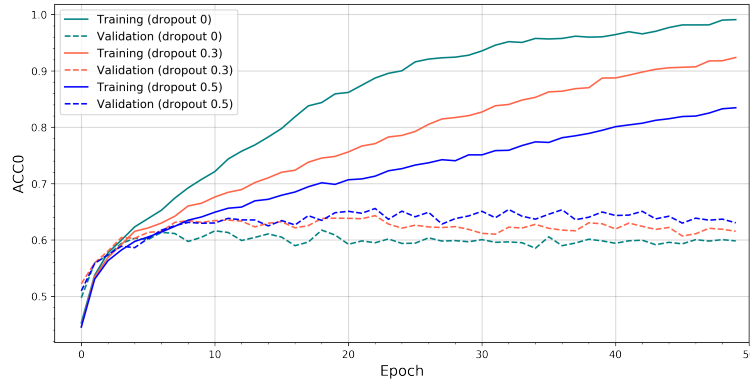


Figure 3.5: Training and validation curves with different dropout probabilities.

Impact of level of details To examine how the different levels of details from bridge inspection reports can impact the condition rating performance, this section first defines a measure of level of details (LoD). Considering that 1) each sentence in the inspection reports typically describes a local condition detail as identified during field inspections and 2) bridges with poorer conditions typically correspond with more sentences of narrative descriptions, the LoD is computed using the scaled number of sentences in each inspection report. To compute the scaled number of sentences, the median m_k and Interquartile Range IQR_k for the number of sentences were first computed for

each condition category, where the IQR_k is calculated as the distance between the third and first quantiles ($Q3 - Q1$). The number of sentences L_j from an inspection report with condition rating k is then scaled using Equation 3.26 to compute the LoD measure.

$$LoD = \frac{L_j - m_k}{IQR_k} \quad (3.26)$$

Figure 3.6 illustrates the histogram of the obtained measure of detail levels for the testing data. Sorting the testing data by ascending level of detail and dividing them into five bins with equal numbers of testing data points, Table 3.5 presents the condition rating performance of the testing data in the five resulting bins. An improvement of condition rating performance can be identified for detail levels around zero ($[-0.167, 0.167]$), which correspond to inspection reports with the number of sentences around the median of its condition category. The condition rating performance drops at both the high and low ends of detail levels, which indicates that anomalies in the level of detail in the form of unusual length (either too much or too little) both harm the rating performance. It should also be noted that, while the distribution of detail levels has a long tail towards the high end, the condition rating performance did not drop as significantly as compared to the low end. This demonstrates that missing details introduce more harm to the rating performance than unusually excessive details.

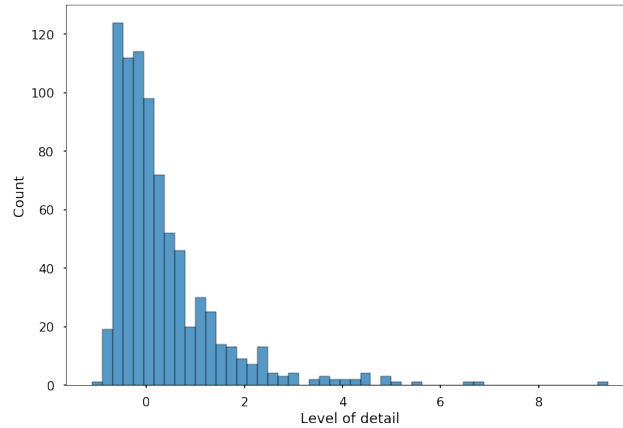


Figure 3.6: Histogram of the level of detail (LoD) measure for the testing data.

3.6.2 Interpretation of Attention Weights

For a closer examination of the hierarchical attention mechanism, this section visualizes the heat map of word attention weights and sentence attention weights using example paragraphs of bridge

Table 3.5: Condition rating performance with different levels of details in the inspection reports.

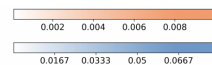
Detail Level		ACC0 (%)	ACC1 (%)	MSE	MAE
Min	Max				
-1.107	-0.435	63.75	98.13	0.419	0.381
-0.435	-0.167	71.25	98.75	0.325	0.300
-0.167	0.167	75.00	99.38	0.269	0.256
0.167	0.833	70.63	98.13	0.350	0.313
0.833	5.5	67.50	99.38	0.344	0.331

deck condition descriptions (Figures 3.7, 3.8, 3.9). Each row displays a sentence (some long sentences may wrap around the next row). The blue color in the left column represents the sentence’s attention weight. The orange color of each word represents a normalized word attention weight. The deeper color indicates higher attention weight. As outlined before, the word attention weights in each sentence and the sentence attention weights in each document were computed separately as presented in Equation 3.7, 3.12. To enable the comparison of word attention weights across sentences, the normalized word attention is computed as $\sqrt{\alpha_s} \alpha_w$, where α_w and α_s denote word and sentence attention weights, respectively. Normalizing α_w by the square root of α_s ensures that the high-weight words in high-weight sentences are correspondingly emphasized in the visualization, and the high-weight words in low-weight sentences are still not completely weighted out.

Figure 3.7 presents two examples of bridge deck descriptions from the testing set that were correctly assigned a condition rating by the proposed model. Figure 3.7 (a) shows an example description of a “Good” bridge deck with a condition rating of 7. The model paid higher attention to the words such as “good condition”, “light”, “minor”, “moderate”, and “satisfactory condition”, which all indicate good conditions of the bridge deck. The model also highlighted the word “joint”, “expansion joint”, which demonstrated an interesting insight as joint deterioration and leakage is usually an early cause of more advanced defects. The presence of minor joint problems in the absence of other major defects can be an indication of earlier stages of overall bridge deterioration consistent with a rating of 7 [176]. The sentences describing pier edge spalling are also highlighted, which describes the prominent defect in this example bridge deck. By aggregating the weighted information of the words and sentences, the model assigned a condition rating of 7, which is defined as “good condition with some minor problems” by the FHWA Coding Guideline [1]. Figure 3.7

good the concrete deck is in overall good condition
 there is light stone and debris accumulation along the right shoulder
 see the expansion joints section for notes regarding the joint at pier 6 that does not have joint armor
 good the exposed concrete deck soffit in spans 1 3 is in good condition
 there are no notable defects
 stay in place sip metal forms are in all bays in spans 4 7
 the sip forms are in good condition throughout
 the deck overhang soffits are in good condition in all spans
 good the concrete parapets are in overall good condition in all spans
 there are typical hairline vertical cracks some with moisture staining in the parapets spaced every 4 to 8 feet
 minor isolated horizontal hairline cracks were also observed at some locations see photo 1
 the utility junction box covers are in good condition
 all fastening screws are in place and seated
 the outside faces of the parapets are in good condition with no notable defects
 good the scupper inlet and downspout assemblies and hardware are in overall good condition
 the neoprene drainage troughs and downspout anchor assemblies are in good condition
 there is light debris accumulation in the span 4 scupper near pier 3
 the scupper is functioning as intended and the downspout is clear
 the drainage trough at pier 7 is leaking from the left end directly onto the end of the pier cap
 the drainage trough has a moderate accumulation of debris which may be clogging the drainage downspout
 not visible inside parapet the utility junction box covers are in good condition with all fastener screws installed and tightened
 good all components of the steel plate unknown joint assemblies and anchor hardware at piers 7 to 1 and 7 2 are in good condition with no notable defects see photo 2
 the compression and strip seal joints are in overall good condition at pier 3 pier 6 and the approach slab
 at pier 3 there is heavy debris accumulation along the full length of the joint
 see photo 3
 the seal is completely obscured from view
 the steel joint armor is in satisfactory condition with light wear and minor surface corrosion
 at pier 6 there is intermittent edge spalling up to 1 inch wide along both edges of the joint with some gaps and separation of the seal see photo 4
 there are minor 0 inch wide longitudinal cracks in the deck on both sides of the joint
 at the time of inspection active water leakage was observed at the joint during rain causing drainage down the front face of the cap in bays 1 and 3

Condition rating 7, predicted as 7

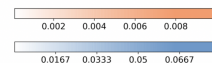


(a) Example descriptions of a “Good” bridge deck with the condition rating correctly predicted as

7.

condition of asphalt similar to last inspection
 original asphalt cracked full width over all timber planks
 photo 1 two asphalt patched areas near abutment a approximately 200 square foot and one asphalt patch near abutment b approximately 100 square foot are broken deteriorated and failing
 2 square foot pothole in patch located approximately 9 feet from abutment b was repaired with asphalt
 1 square foot pothole up to 1 inch deep in patched area at abutment a was repaired with asphalt
 both repair patches are cracked and beginning to fail
 photo 2 in previous inspections a 2 square foot pothole with a failed 4 inch wide 16 inch long section of timber decking below pothole was noted at abutment b
 this failed area was not visible during this inspection and the pothole was patched with asphalt
 condition of deck similar to last inspection
 14 timbers adjacent to abutment a are deteriorated and move under traffic loads
 6 timbers adjacent to abutment b are deteriorated and move under traffic loads
 moisture present on all timbers at edges of deck
 light to medium dirt and debris along both curb lines
 photo 3 light vegetation growing along both curb lines photo 3 condition of deck similar to last inspection
 moisture seepage present on all timbers at edges of deck
 deck bolts throughout structure are loose and rattle under traffic loads
 two areas near abutment b have deck timbers with failed missing sections photo 4 1 square foot at 6 foot from face of abutment b between beams 4 and 5
 5 square foot at 6 foot from face of abutment b between beams 8 and 9
 extensive deterioration such as splitting checking and decay noted in both timber curbs
 loose bolts noted throughout curbs
 timber curb at upstream end of abutment a has completely failed and is missing
 photo 3 75 percent of upstream curb exhibits delay and is beginning to fail
 photo 3 4 6 timber curb supports at both curbs have failed or are missing in 4 out of 7 locations on downstream side of structure
 guardrails used as bridge railings
 damage to top of guardrail at both sides of structure
 photo 3 anchor bolts for guardrail attachment are pulling thru timber curb
 railing is loose at both sides of structure

Condition rating 4, predicted as 4



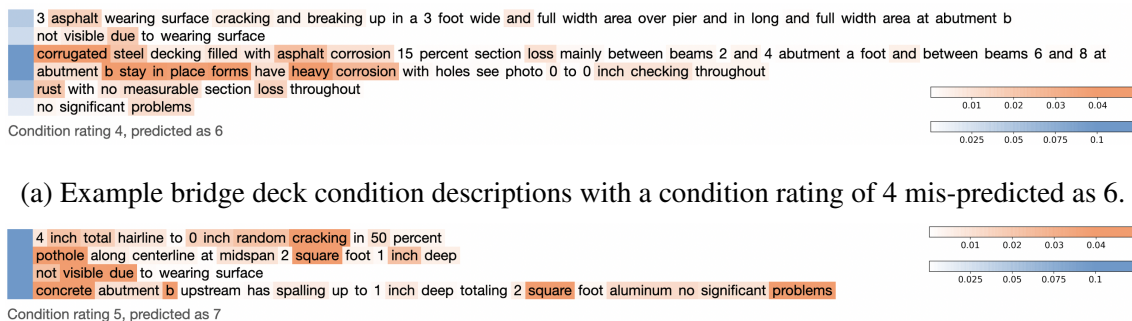
(b) Example descriptions of a “Poor” bridge deck with the condition rating correctly predicted as

4.

Figure 3.7: Example descriptions from the testing set that were correctly assigned a condition rating by the proposed model.

(b) shows an example description of a “Poor” bridge deck with a condition rating of 4. The model highlighted words such as “broken”, “deteriorated”, “failing (failed)”, and “decay”, which are indicators of poor conditions. As shown in the blue column of sentence attention weights, the model emphasized in sentences that describe “pothole”, “timbers deteriorated”, “moisture on timbers”, “deck bolts loose”, and “extensive deterioration such as splitting and decay”, which all describe prominent deterioration of the described timber bridge deck. The non-structural elements such as curbs, guardrails, and railings described towards the end of this example were not emphasized by the model.

As for mispredicted cases, as presented in the previous section, only 1.25% of deck descriptions in the testing set were mispredicted by more than one level. Examination of these cases revealed that a number of these errors can be attributed to inadequately short description of some bridge decks with Poor or Fair condition ratings. Figure 3.8 presents two examples of short descriptions of bridge decks of condition ratings 4 and 5 that were mispredicted by the model as conditions 6 and 7, respectively. Although both examples have sentences describing some severe damages such as “heavy corrosion” and “pothole along centerline”, it is hard to judge based on the descriptions that the bridge deck is in condition 4 or 5 even via human examination. Although both the federal and state inspection guidelines [2, 24, 25] recommend documenting every detail of local conditions especially for bridges with Poor or Fair condition, it is anticipated that further standardization of inspection note-taking procedures, *e.g.* documenting each individual local deficiencies with one sentence, would further improve the performance of the automatic condition rating model. Figure 3.9 shows another off-by-2 error case, which is part of a long description of a bridge deck with condition 6 that is mispredicted as condition 4.



(b) Example bridge deck condition descriptions with a condition rating of 5 mis-predicted as 7.

Figure 3.8: Example incomprehensive descriptions mis-predicted as two levels higher.

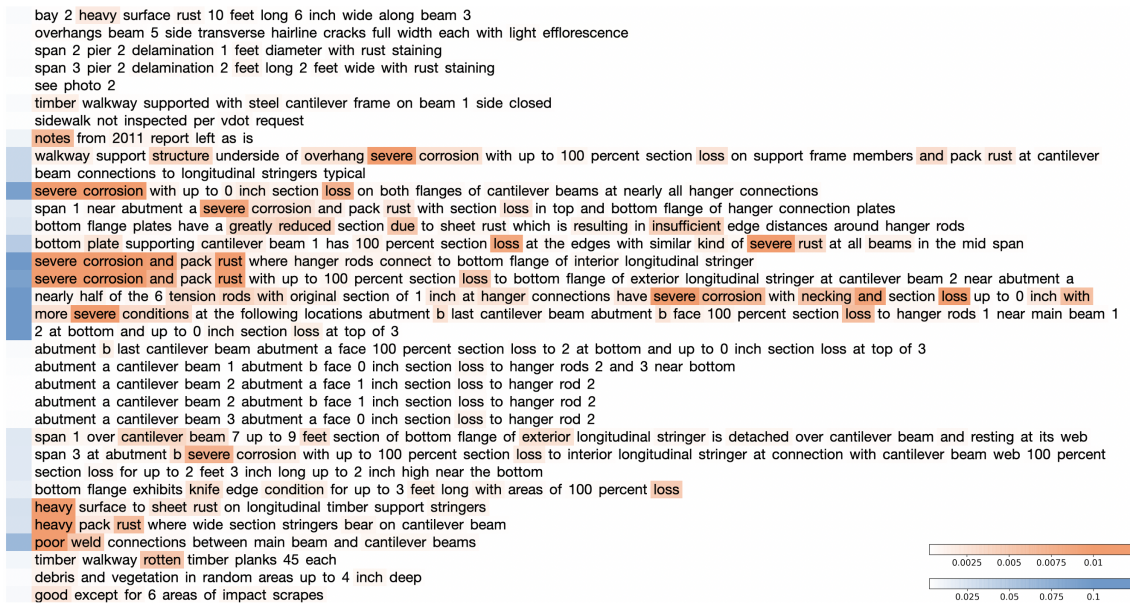


Figure 3.9: Example bridge deck condition descriptions with a condition rating of 6 but mis-predicted as 4.

Focusing beyond individual attention examples, Table 3.6 analyzes the most frequent top-weighted words for descriptions of different condition ratings. The table presents a column of top-weighted words and the next column of the words' frequencies divided by the number of documents for each condition. Since the word attention weights are the probability scores computed separately within each sentence, the values of these word attention weights cannot be directly compared across sentences; similarly, the sentence attention weights cannot be directly compared across inspection documents. Therefore, the top-weighted words were generated by selecting two top-weighted sentences from each bridge deck description, and then selecting three top-weighted words from each of the two sentences. As presented in Table 3.6, the top-weighted words in conditions 4 and 5 were mostly words of deficiencies such as “delaminated”, “decay”, “spalled”, and “scale”. The word “loss” was also among the top-weighted words for conditions 4 and 5, which is commonly used to describe the severity of reinforcement corrosion in section loss. The word “joint” tends to be emphasized in condition 6, 7, and 8, which suggest that defects associated with bridge joints might be prominent in these three condition levels. Condition 7 and 8 contain top-weighted words such as “hairline”, “scattered”. “random”, and “isolated”, which indicate the emerging phase of local deficiencies such as hairline cracks and scattered spalling. The top words in condition 8 and condition 9 were mostly words describing good conditions such as “good”, “no significant problems”, “no deficiencies noted”. On the other hand, the top defects in conditions 4 and 5 are “delamina-

Table 3.6: List of 20 top-weighted words for different condition ratings.

Condition 4		Condition 5		Condition 6		Condition 7		Condition 8		Condition 9	
Word	Freq	Word	Freq	Word	Freq	Word	Freq	Word	Freq	Word	Freq
delaminated	0.318	delaminated	0.327	concrete	0.204	cracking	0.217	no	0.389	good	0.71
decay	0.265	concrete	0.241	cracking	0.193	hairline	0.204	problems	0.325	no	0.618
spalled	0.203	delamination	0.212	delaminated	0.159	good	0.154	significant	0.26	problems	0.42
concrete	0.189	spalled	0.17	hairline	0.129	concrete	0.1	concrete	0.209	significant	0.374
delamination	0.178	cracking	0.137	delamination	0.109	light	0.089	good	0.194	deficiencies	0.145
loss	0.12	scale	0.13	fair	0.106	epoxy	0.088	asphalt	0.186	noted	0.099
scale	0.106	decay	0.114	light	0.103	stay	0.078	cracking	0.115	concrete	0.076
deterioration	0.095	light	0.09	scale	0.088	joint	0.064	scattered	0.094	sip	0.069
cracking	0.092	fair	0.089	epoxy	0.086	scattered	0.061	deficiencies	0.085	not	0.069
decayed	0.072	loss	0.089	spalled	0.079	asphalt	0.058	hairline	0.067	asphalt	0.069
severe	0.058	areas	0.078	good	0.069	random	0.054	light	0.062	condition	0.069
timbers	0.053	due	0.069	due	0.068	transverse	0.054	thick	0.059	slab	0.061
spall	0.053	spall	0.064	joint	0.067	isolated	0.053	except	0.051	notes	0.061
heavy	0.047	heavy	0.051	long	0.061	joints	0.052	wearing	0.046	slabs	0.053
due	0.047	scattered	0.046	stay	0.059	long	0.051	joint	0.043	forms	0.053
moisture	0.045	efflorescence	0.045	isolated	0.058	no	0.049	planks	0.042	found	0.046
areas	0.045	spalling	0.045	decay	0.057	problems	0.044	prestressed	0.042	hairline	0.038
soft	0.042	long	0.045	overlay	0.056	spall	0.042	joints	0.042	superstructure	0.038
spalling	0.042	linear	0.045	areas	0.047	overlay	0.042	epoxy	0.04	noteworthy	0.038
floor	0.042	epoxy	0.042	linear	0.047	efflorescence	0.04	stay	0.04	except	0.031

tion” and “spalling” which usually occur in the later stages of deterioration after hairline cracks grow into larger separations and discontinuities in material resulting in area or volume defects. Accordingly, the position of delamination and spalling moves toward the end of the list in condition 6 and 7, replacing with cracking (especially hairline) as the top defect word in condition 7 and 6. This is in agreement with the progression of minor cracks into more serious deficiencies in bridges having worse condition. Another notable observation is the position of adjectives of different strengths among the different conditions. Specifically, higher intensity adjectives such as “severe” and “heavy” were among the top-weighted words in conditions of 4 and 5. The medium intensity adjective “fair” were emphasized in conditions 6 and 5, while “scattered” peaks at 7 and 8 and “light” appears in the top-weighted words across conditions 5 to 8. The adjective “hairline” that is usually used to describe the onset of cracking, is most frequently emphasized in condition 7 and with a diminishing frequency in the condition states immediately next to 7. It should be noted that the adjective “significant” that appears at the top of conditions 8 and 9 is a part of the phrase

”no significant problems”.

3.6.3 Quality Control Performance

In addition to the application of automatic condition rating, the proposed framework also supports quality control of ratings assigned by in-training and potentially less experienced inspectors. The quality control tool processes the narrative descriptions together with the inspector-assigned rating and generate a recommendation of whether the assigned rating is in alignment with the data. The recommendation is generated by computing a Likelihood Ratio (Equation 3.22) and comparing it with a data-driven threshold Θ . As explained in Section 3.4.4, while a direct quality control approach can screen the inspector-assigned ratings by directly comparing with the automatic condition rating model outputs, the proposed quality control process develops a data-driven threshold that recommends to accept/reject the inspector-assigned ratings with a unified level of certainty. The tuning of the accept/reject threshold helps generate more accurate quality control recommendations. This section presents the performance of the proposed quality control compared to the direct method in terms of the evaluation metrics as defined in Equations 3.23, 3.24, and 3.25. For this evaluation, a synthetic set of “inspector-assigned” ratings were generated, where 50% of the condition ratings were correctly assigned (the same as the condition rating ground truth) while the other 50% were randomly sampled from a binomial distribution of $\{r, r \in [4, 9] \text{ and } r \neq \text{ground truth rating}\}$. The hyper-parameter Θ was selected based on the performance of the F1 score computed in a 5-fold cross validation using the training data. Table 3.7 presents the performance of the proposed quality control using the likelihood ratio presented in Section 3.4.4 and the direct quality control. In the direct quality control, the system accepts the provided rating when it equals the model output prediction and rejects otherwise. Based on this table, the proposed quality control demonstrated significant improvement in overall accuracy and F1 score of the “accept/reject” recommendations. Although HAN achieved a high precision of 92.56%, the recall of HAN is below 70%; the proposed method achieved above 90% recall with 86.2% precision.

Table 3.7: Quality control performance compared with the baseline system.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Direct Comparison	81.57	92.56	69.59	79.44
Likelihood Ratio	88.67	86.20	92.70	89.33

To examine the number of data points needed for tuning the hyper-parameter Θ , Figure 3.10 presents the quality control performance when different numbers of data points were used for tuning Θ . Different sets of data with varying sizes were randomly sampled from the entire validation set for the tuning of Θ , and in each sample, the Θ value with the optimal F1 score was selected for performance evaluation using the testing set. The experiment with each sample size was repeated 50 times, producing a distribution of quality control performance in terms of precision, recall, and F1 score. Figure 3.10 then plots the *mean* performance together with the range of *mean \pm standard deviation* performance for the three metrics at each number of data points. It can be seen from the figure that the quality control performance improves as the sample size increases, and meanwhile, the variance of performance also decreases, indicating a more reliable performance. Given the stable convergence of the F1 score above 88% after 200 data points, this sample size was considered an appropriate number required for this specific dataset for tuning the hyper-parameter Θ for a relatively reliable quality control performance.

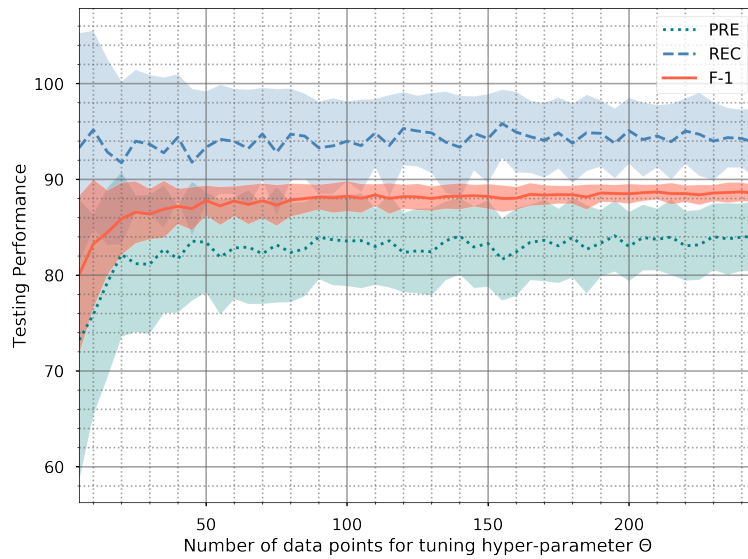


Figure 3.10: Quality control performance with different number of data points used for tuning parameter Θ

3.6.4 Analysis of Quality Control Scenarios

This section discusses how different types of errors and bias in the inspector-provided ratings can affect the model's quality control performance. Although in reality the inspector-provided condition ratings might contain a combination of different types of errors, to study how well the model can

detect each of type of error, this study generated five assumed error scenarios. Table 3.8 presents the five error scenarios studied and the generation approach for the wrong ratings of each error type. For all five scenarios, 50% of the condition ratings were incorrect ratings generated based on the selected scenario, and the other 50% were correctly assigned (the same as the ground truth) to allow for the evaluation of both negative and positive decisions. The effect of the percentage of wrong ratings is also examined and discussed later in this section.

Table 3.8: Five assumed error scenarios and the approach for generating the incorrect ratings of each error type.

Scenario Name	Generation of the Error Ratings
Rand	Randomly sampled from binomial distribution: $\{r, r \in [4, 9] \text{ and } r \neq \text{ground truth rating}\}$
UpOne	One level higher than the ground truth
UpTwo	Two levels higher than the ground truth
DownOne	One level lower than the ground truth
DownTwo	Two levels lower than the ground truth

Figure 3.11 presents the quality control performance of accepting or rejecting the provided ratings in the five error scenarios. As can be seen in this figure, the model achieved the highest performance in the UpTwo and DownTwo scenarios, with all four evaluation metrics over 90% (except UpTwo precision which is slightly below 90%). The UpOne and DownOne scenarios are relatively

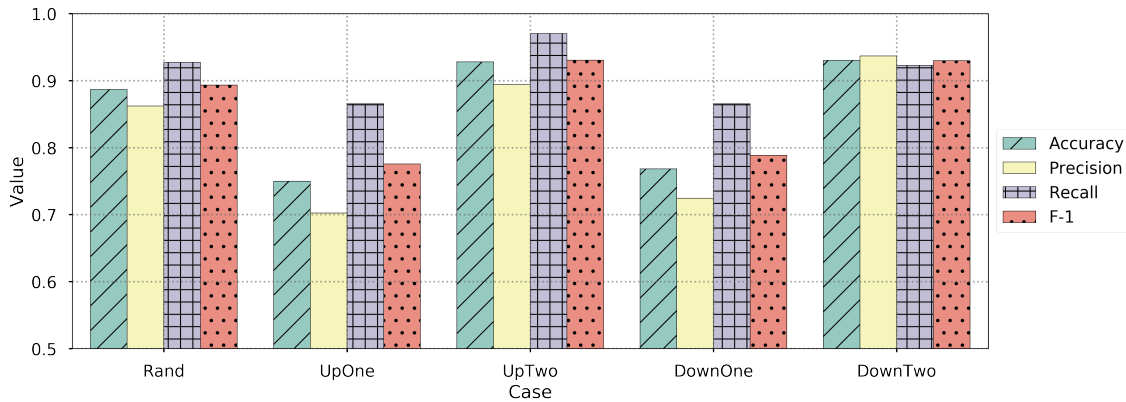


Figure 3.11: Quality control performance in five error scenarios.

harder for the model to decide, and the lower precision and higher recall in these two scenarios indicates that the false decisions include more false positives (failing to identify wrong provided ratings) than false negatives (failing to identify correct provided ratings), which highlights the difficulty in differentiating between adjacent categories. The random scenario (Rand) demonstrated better performance than the two off-by-one cases but is not as strong as the two off-by-two cases.

The hyper-parameter Θ plays an important role in the decision of accepting or rejecting the provided ratings. For each assumed rating scenario, the effect of the hyper-parameter Θ on the quality control performance was studied and presented in Figures 3.12 and 3.13. It should be noted that in each case Θ was tuned using cross validation and the quality control performance was reported using the testing set. The following analysis presents a summary of the factors influencing Θ , and provides insights for tuning Θ based on engineering knowledge and the application scenario:

1. The value of Θ can be affected by the type of errors in the provided ratings. Figure 3.12 illustrates the variation of F1 score with the value of Θ in the five error scenarios. For parameter tuning in each error scenario, five-fold cross validation was performed where the training data were split into five folds, four of which were used to train the model, and the remainder was used to compute the F1 scores as Θ varies. Figure 3.12 shows that each scenario has a slightly different optimal Θ (denoted by the peak of the curves), and that the two off-by-two scenarios (UpTwo and DownTwo) performed better than the two off-by-one scenarios (UpOne and DownOne) indicating that adjacent categories are relatively harder to differentiate. Accordingly, in the off-by-two scenarios, the optimal value for Θ is smaller than the off-by-one scenarios, representing a lower threshold to accept and higher trust in the model-generated

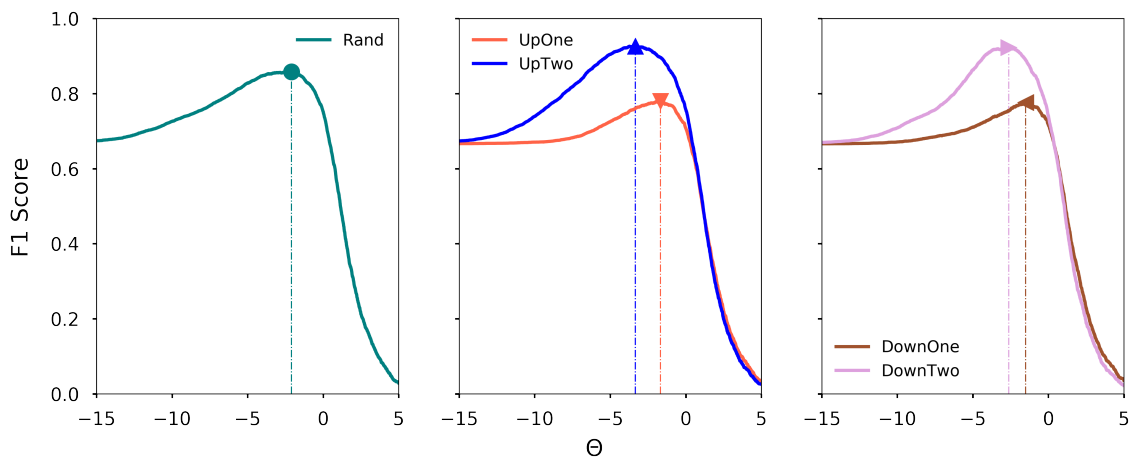


Figure 3.12: Variation of F1 scores with the value of Θ in the five error scenarios.

probability scores.

2. The choice of Θ value also depends on which type of error (false positive or false negative) is deemed less desirable in the specific quality control application. False-positive predictions fail to identify incorrectly-assigned ratings, while false negative predictions fail to accept correct ratings. Considering the Rand scenario as an example, Figure 3.13 illustrates the trade-off between the two types of errors in selecting an optimal value for Θ . As the value of Θ increases, the percentage of type I error (false positive) decreases while the percentage of type II error (false negative) increases. Accordingly, similar trends appear in the variation of recall and precision with changes in Θ . Although the value of Θ can be selected based on F1 score to include the simultaneous effect of recall and precision, bridge managers can choose to prioritize a certain error type at the cost of the other based on the owner's priorities and expected outcomes.

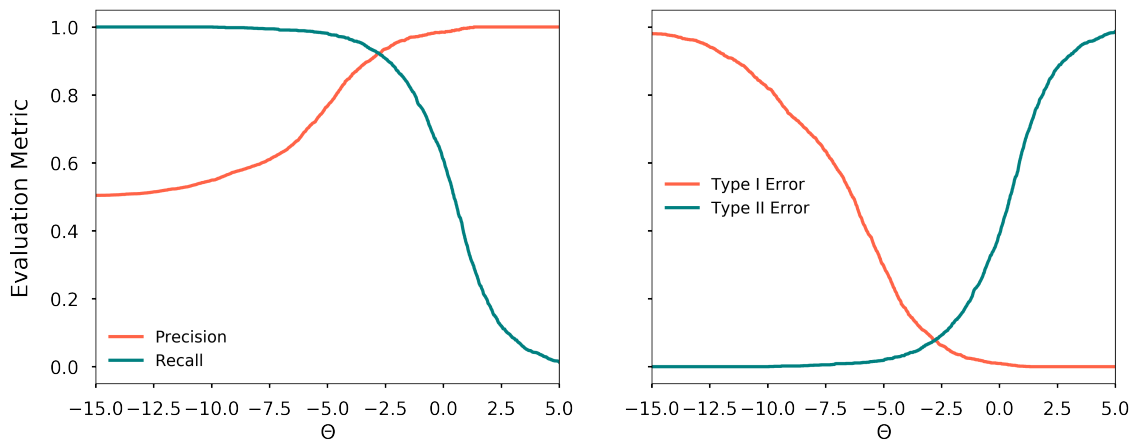


Figure 3.13: Trade-off between precision and recall (left), and trade-off between the false positive or false negative errors (right) in selecting an optimal value for Θ . (*Figure generated in Rand scenario.)

3. Prior knowledge regarding the confidence in the provided ratings can also affect the value of Θ . Figure 3.14 illustrates the optimal Θ values and the associated quality control accuracy with different percentages of incorrect ratings in each error scenario. It can be seen that, all five error scenarios demonstrated an increasing trend in the optimal Θ value as the percentage of incorrect rating increases. Therefore, a higher Θ would be more appropriate if higher levels of error were anticipated in the provided condition ratings (e.g., inexperienced inspectors

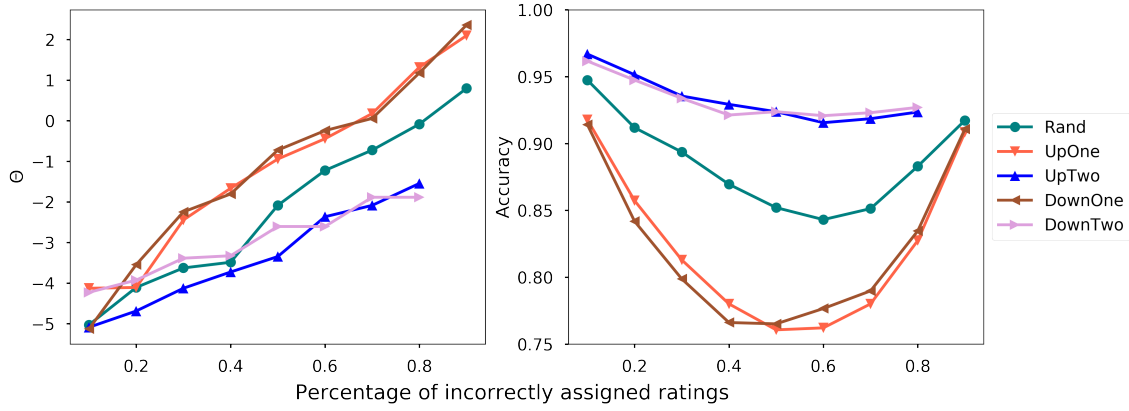


Figure 3.14: The optimal Θ values (left) and model accuracy (right) for different percentages of wrong ratings. (*UpTwo and DownTwo cases do not include 0.9 datapoints because not enough ratings can be shifted up or down by two levels.)

under training). It can also be seen in Figure 3.14(right) that the model achieved higher accuracy at both high and low percentages of incorrect ratings than the middle of the range (40%-60% of the ratings). This is because the model can rely on a lower threshold (tend to accept) in the case of low percentage of incorrect ratings and a higher threshold (tend to reject) in the case of high percentage of incorrect ratings, but the model relies more on accurate model-generated probability scores in deciding to accept or reject when the number of correct and incorrect provided ratings are close.

Chapter 4

Fusing Visual and Textual Representations from Bridge Inspection Reports for Reliable Automated Condition Assessment

Li, T., and D.K. Harris, Fusing Visual and Textual Representations for Automated Condition Assessment. To be submitted to Journal of Civil Structural Health Monitoring, 2021.

4.1 Abstract

This study identifies bridge inspection reports and the associated historical condition ratings as a collective knowledge base of bridge condition assessment, and proposes a deep learning-based fusion approach for automated bridge condition rating using the visual and textual data from bridge inspection reports. Considering the structure of inspection reports that each contains a collection of images and a sequence of sentences that document local bridge conditions, the proposed fusion approach constructs visual and textual representations from images and sentences separately, and adopts a sequence encoder followed by an attention mechanism to fuse multi-modal representations to support condition rating. While the image-based defect recognition and condition assessment models have been extensively studied in the existing literature, results from this study show that the visual modality alone did not yield satisfactory condition rating performance. Condition rating using textual data from the inspection reports significantly outperformed the visual modality, and the proposed fusion approach introduced further improvements over the uni-modal baselines. This

study further investigated the uncertainty of rating predictions under random disturbance introduced by data augmentation and dropout training strategy. The uncertainty analysis showed that 95% of the rating predictions for the testing data vary within 0.535, and referring the uncertain predictions to human investigations can further improve the rating performance. The proposed model can be used to process the bridge condition data collected from the current visual inspection practices to improve rating consistency, and discussions of this study points to the potential improvement in future inspection data collection that can further facilitate automated condition assessment.

4.2 Introduction

Towards addressing the critical preservation challenges of bridge infrastructure and the challenges in obtaining consistent and reliable condition assessment, the increasing volume of multi-modal bridge condition data offers a viable approach that supports the integration of efficient automation concepts. As visual field inspection remains the predominant approach to collect the conditions of bridges, a significant number of recent works have developed robotic systems using unmanned aerial or ground vehicles (UAV/UGV) [28, 29] that navigate around bridges and collect visual data [30–33]. As of 2018, the Department of Transportation (DOT) in 15 states are actively conducting researches on the use of UAVs in inspection, 20 states have incorporated UAVs into their daily operations [34]. The robotic inspection systems demonstrate great potential in alleviating labor cost and safety concerns in visual inspection, and more importantly, it produces an increasing volume of data for supporting the automation in the subsequent condition assessment process. Extensive research efforts have been made in vision-based automated condition assessment. The scale of assessment ranges from image/patch-level defect classification [61, 62], object-level defect detection [191], as well as pixel-level defect segmentation [45, 63]. However, it requires further research efforts in order to progress towards quantifying local deficiencies and aggregating them to global condition ratings.

Besides collecting automated inspection data using robotic systems, the bridge infrastructure system is also historically rich in both structured tabular data such as the National Bridge Inventory (NBI) [1] and unstructured descriptive data such as inspection reports and maintenance records. The NBI contains characteristics (*e.g.* geometry, structural systems, materials, etc.), as well as condition rating scores assigned through visual inspections, for more than 616,000 bridges in the United States (US) since 1992. Besides, bridge inspection reports document the condition evolution of the entire bridge population in richer details to the extent of every defect identified in field inspection, the

expert condition assessment (rating scores and recommendations), as well as the follow-up maintenance activities. These inspection reports and the associated condition rating scores collected over the years represent a knowledge base that aggregates individual inspectors' expertise, and can shed light on how local conditions should be mapped to the rating scores. Automated information extraction approaches have been developed to extract condition-related information from unstructured textual descriptions from the inspection reports [39, 40, 162]. An automated condition rating model has also been developed that map the textual descriptions from inspection reports to the condition ratings [164]. The extracted condition descriptions, together with the NBI data, have been used for enhanced deterioration prediction [163].

While visual and textual bridge condition data and automated condition assessment tools have become increasingly available, the problem of fusing both modalities for automated condition assessment has not been extensively researched. Survey studies have been conducted in the field of multi-modal analysis [59, 141], and most of the collected works suggested the superiority of multi-modal over uni-modal approaches. A large amount of research has developed models for visual and textual fusion using large-scale data with general contents [136–138], while their applicability to the domain of bridge condition assessment remains unknown due to the unique content and structure of bridge inspection report data. Table 4.1 illustrates the visual data (images) and textual data (descriptive sentences) for bridge deck from a typical inspection report. The descriptive sentences state in detail every local defect identified during inspections, while the images serve as supplementary information that illustrates zoomed views of selected local defects. Each image and sentence correspond to one inspection finding, typically a local defect condition without the direct alignment between each image-sentence pair. The color-coding of the sentences and images indicates the alignment between the two modalities. One image may correspond to the content of multiple sentences (Image 1 - sentence 1,2; Image 2 - sentence 7-9), or the other way around (Sentence 12 - image 3-5). Sentences may also not correspond to any images, as shown in black in Table 4.1. While the existing fusion approaches were developed for input modalities that have perfect alignment and contain complementary information [59], fusing such loosely-aligned modalities with supplementary information in this practical use case of visual and textual data from the inspection reports is still challenging and warrants further study.

To that end, this study constructs visual and textual representations for images and sentences separately, and proposes a Recurrent Neural Network (RNN) sequence encoder and an attention

Table 4.1: Visual data (images) and textual data (descriptive sentences) for bridge deck from a typical inspection report.






1	Asphalt overlay typically has several transverse cracks at expansion joints, up to 1" wide x full joint length.	
2	Pier 1: Northbound lane wearing surface is breaking up over 1' width over joint, with 2 SF full depth pothole.	
3	Pier 6: Southbound lane wearing surface is breaking up 6" wide over joint.	
4	Not visible below wearing surface except at the concrete-armored expansion joints over each abutment where moderate spalling and cracking is evident.	
5	Edge spalling along deck joints typically 12" long x 3" wide x 1" deep.	
6	Several edge spalls have been patched with asphalt in the past.	
7	Span A NBL adjacent to Abutment A joint: Spall, 4' long x 1' wide x up to 4" deep with exposed reinforcing.	
8	Spall has been filled with asphalt in the past.	
9	Span A NBL, both lanes adjacent to Abutment A joint: Spalling, 10' long total x up to 4" wide x 2" deep.	
10	Map cracking, moisture seepage, efflorescence (typical throughout approximately 10% of deck area) and smoke staining (over tracks).	
11	Underside of deck has isolated areas of longitudinal and transverse cracking up to 1/16" wide on approximately 10% of deck area.	
12	Numerous areas of spalling and delamination throughout underside of deck, totaling 5761 SF of delamination and 256 SF of spalling up to 2" deep with exposed reinforcing.	
13	Cover spalls throughout with exposed reinforcing chairs spaced approximately 6" on center, typically up to 1" diameter x 1/4" deep with exposed reinforcing.	

Image 1

Image 2

Image 3

Image 4

Image 5

mechanism to fuse and summarize the multi-modal representations for predicting an overall condition rating. To evaluate the uncertainty of model predictions, test-time augmentation and dropout is adopted that incorporates randomness during testing and generates a distribution of model predictions whose standard deviation can be used as the measure of model uncertainty. The evaluation of condition rating performance demonstrated the capability of each modality (visual and textual) in supporting condition assessment, and promoted the fusion of the two modalities for enhanced rating performance. Results of the uncertainty analysis showed that the rating performance is relatively reliable compared with the variance estimated in the human-assigned ratings; and referring uncertain rating predictions to human investigation can further improve the rating performance. Discussions of this study also point to potential improvements in future condition data collection process that may further facilitate better automated condition rating models.

4.3 Proposed Multi-modal Rating Model

Given a bridge inspection report that contains a collection of images $\mathcal{V}_k, k \in [K]$ and a sequence of sentences $\mathcal{S}_j, j \in [J]$ ($[x]$ denotes the set of non-negative integers smaller than x ; K and J denote the numbers of images and sentences, respectively), this study extracts visual and textual representations from images and sentences, and fuses the two modalities to predict the condition rating. Figure 4.1 provides an overview of the model architecture that includes an image module and a word module to extract visual and textual representations from the inspection reports, respectively, and a fusion module to combine the extracted representations for condition rating.

The visual representation is extracted from each image using the convolutional dense blocks from the DenseNet [118], which is a well-known convolutional neural network that has been pre-trained using the ImageNet data [119]. Each image \mathcal{V}_k is then represented by a vector $r_k \in \mathbb{R}^{2208}$

$$r_k = \text{DenseNet}(\mathcal{V}_k) \quad (4.1)$$

The vectors $r_k, k \in [K]$ are then condensed to a dimension m , which is the selected common dimension for the visual and textual representation, using a Linear layer,

$$v_k = W^{\mathcal{V}} r_k + b^{\mathcal{V}} \quad (4.2)$$

where $W^{\mathcal{V}} \in \mathbb{R}^{m \times 2208}$, and $b^{\mathcal{V}} \in \mathbb{R}^m$ are the corresponding weight matrix and bias parameters.

The textual representation is constructed using a word embedding layer and a word module as in Hierarchical Attention Network (HAN) [164, 192]. The word embedding layer contains the

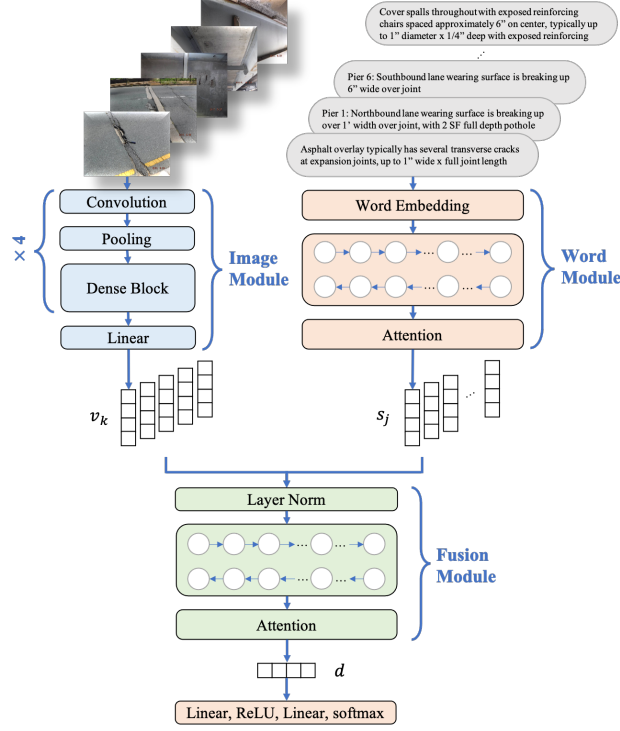


Figure 4.1: Model architecture for the extraction and fusion of the visual and textual representations.

pre-trained Global Vectors (GloVe) [71] that were developed using the general-content Wikipedia data and cover a vocabulary of 84M words. Words from the inspection reports are embedded in a vocabulary look-up manner into their embeddings, and words that are out of the vocabulary of the pre-trained GloVe were embedded with randomly initialized word embeddings using the He Initialization [193]. For a sentence \mathcal{S}_j that contains T_j words, each word $\mathcal{W}_{jt} \in [T_j]$ is then represented by a word vector w_{jt}

$$w_{jt} = \text{WordEmbedding}(\mathcal{W}_{jt}) \quad (4.3)$$

The word module contains a sequence encoder followed by an attention mechanism to aggregate the word vectors $w_{jt}, t \in [T_j]$ into a sentence vector s_j . The sequence encoder contains the bi-directional Gated Recurrent Units (GRUs) [114] that encodes the contextual information from both before and after each word vector into its hidden representation $h_{jt}^{\mathcal{W}}$.

$$h_{jt}^{\mathcal{W}} = [\overrightarrow{\text{GRU}}(w_{jt}); \overleftarrow{\text{GRU}}(w_{jt})] \quad (4.4)$$

where $h_{jt}^{\mathcal{W}} \in \mathbb{R}^m$. The attention mechanism first feeds $h_{jt}^{\mathcal{W}}$ through a Linear layer

$$v_{jt}^{\mathcal{W}} = \tanh(W^{\mathcal{W}} h_{jt}^{\mathcal{W}} + b^{\mathcal{W}}) \quad (4.5)$$

and then calculates the attention weight for each word w_{jt} is via a softmax function

$$\alpha_{jt}^{\mathcal{W}} = \frac{\exp(\mathbf{v}_{jt}^{\mathcal{W}T} \mathbf{u}^{\mathcal{W}})}{\sum_{t'=0}^{T_j-1} \exp(\mathbf{v}_{jt'}^{\mathcal{W}T} \mathbf{u}^{\mathcal{W}})} \quad (4.6)$$

where the vector $\mathbf{u}^{\mathcal{W}}$ represents an overall informative word, and $\mathbf{v}_{jt}^{\mathcal{W}T} \mathbf{u}^{\mathcal{W}}$ computes the similarity between $\mathbf{v}_{jt}^{\mathcal{W}}$ and $\mathbf{u}^{\mathcal{W}}$ as a measurement of word importance. The vector $\mathbf{u}^{\mathcal{W}}$ is the parameters in the attention mechanism that is learned through model training. The sentence vector s_j is then computed as a weighted sum of the encoded word representations

$$s_j = \sum_{t=0}^{T_j-1} \alpha_{jt}^{\mathcal{W}} h_{jt}^{\mathcal{W}} \quad (4.7)$$

where $s_j \in \mathbb{R}^m$. The word embedding layer with GloVe as well as the word module are obtained from the pre-trained HAN model that also includes a sentence model that summarizes the sentence vectors into a document-level representation for condition rating. The HAN model is also used as the uni-model (textual) baseline for the proposed fusion models in section 4.5.

The image vectors $v_k, k \in [K]$ and the sentence vectors $s_j, j \in [J]$ are then combined and fed through the fusion module that starts with the layer normalization [194, 195]

$$F = \text{LayerNorm}([v_0, \dots, v_{K-1}, s_0, \dots, s_{J-1}]) \quad (4.8)$$

where $\text{LayerNorm}(X) = \frac{X - \mathbb{E}(X)}{\sqrt{\text{Var}(X) + e^{-5}}}$. The fused matrix $F \in \mathbb{R}^{m \times (K+J)}$ contains both visual and textual representations from an inspection report. Each column of F was further re-indexed as $f_l, l \in [L]$, where $L = J + K$.

The fusion module also contains a bi-directional GRU sequence encoder that fuses $f_l, l \in [L]$ into a hidden representation $h_l^{\mathcal{F}}$ via the recurrent connections among the GRUs

$$h_l^{\mathcal{F}} = [\overrightarrow{\text{GRU}}(f_l); \overleftarrow{\text{GRU}}(f_l)] \quad (4.9)$$

The attention mechanism in the fusion module then computes a fusion weight for each hidden representation of the images and sentences

$$\mathbf{v}_l^{\mathcal{F}} = \tanh(W^{\mathcal{F}} h_l^{\mathcal{F}} + b^{\mathcal{F}}) \quad (4.10)$$

$$\alpha_l^{\mathcal{F}} = \frac{\exp(\mathbf{v}_l^{\mathcal{F}T} \mathbf{u}^{\mathcal{F}})}{\sum_{l'=0}^{L-1} \exp(\mathbf{v}_{l'}^{\mathcal{F}T} \mathbf{u}^{\mathcal{F}})} \quad (4.11)$$

where $W^{\mathcal{F}}$ and $b^{\mathcal{F}}$ are the weight matrix and bias in the fusion module; $\mathbf{u}^{\mathcal{F}}$ is a parameter vector optimized during model training as a representation of an overall informative fused vector (image

or sentence). The fused document vector d is then computed as the weighted sum of $h_l^{\mathcal{F}}$

$$d = \sum_{l=0}^{L-1} \alpha_l^{\mathcal{F}} h_l^{\mathcal{F}} \quad (4.12)$$

The document vector is processed by two sequential Linear layers with ReLU as activation functions (denoted as function g) followed by a softmax classifier to be mapped to the a condition rating

$$q = W^{\mathcal{P}} g(d) + b^{\mathcal{P}} \quad (4.13)$$

$$p_c = \frac{\exp(q_c)}{\sum_{c'=0}^{C-1} \exp(q_{c'})} \quad \forall c \in [C] \quad (4.14)$$

where $W^{\mathcal{P}}$ and $b^{\mathcal{P}}$ denote the corresponding weight and bias parameters; $q \in \mathbb{R}^C$ where q_c represents the c_{th} element in q , and C is the total number of condition rating categories. The vector $p \in \mathbb{R}^C$ consists of p_c scores, which computes the probability of the condition being rated as c . Given the target condition rating $y \in \{1, 2, \dots, C\}$, the cross entropy loss used for model training is

$$L(p, y) = -\log(p_y) \quad (4.15)$$

To assign a condition rating, the model computes the predicted rating \hat{y} as

$$\hat{y} = \arg \max_c p_c \quad (4.16)$$

4.4 Data Collection and Preparation

Bridge inspection reports used in this study were collected from the inspection report database maintained by the Virginia Department of Transportation (VDOT). The most recent reports for all bridges in the state of Virginia were downloaded from the database in June 2018. Accessing historical inspection reports of previous inspections from the database requires a manual process that navigates to each bridge and downloads by the year, and therefore, the historical inspection reports were not extensively collected (*i.e.* report history before 2018) in this study due to labor cost. Each inspection report contains separate sections for bridge components such as Deck, Superstructure, and Substructure that document the inspection findings as well as a condition rating of 0-9 assigned by the inspector for each component. This study was constrained to the development of a condition rating model for the deck component; however, similar models could also be constructed for the other bridge components (*i.e.* Superstructure and Substructure). Considering that only 1.64% of the

bridge decks in Virginia were in condition 4 or below as of 2018, 245 additional reports of bridges from other inspection cycles (prior to 2018) that were historically rated as condition 4 or below were manually downloaded from the VDOT database to support this study. Figure 4.2 illustrates the composition of the collected reports in terms of the corresponding VDOT districts, number of spans, structural designs, and materials. The collected inspection reports were submitted by the nine VDOT district offices (as illustrated in Figure 4.2 top), where each district office adopts different teams and personnel for routine bridge inspection, and therefore the collected reports aggregate a wide coverage of inspector’s expertise that can guide the automated condition rating model. Over 90% of the collected reports associate with bridges that contain less than 5 spans. The majority of the bridges (68.5%) were multi-girder bridges, and the next popular structural design was slab, T-beam, and Box-beam, sequentially. Concrete and steel are the two main materials identified in the collected bridges, the rest of other materials (*e.g.* timber, masonry) contributed to 0.5% of the collection.

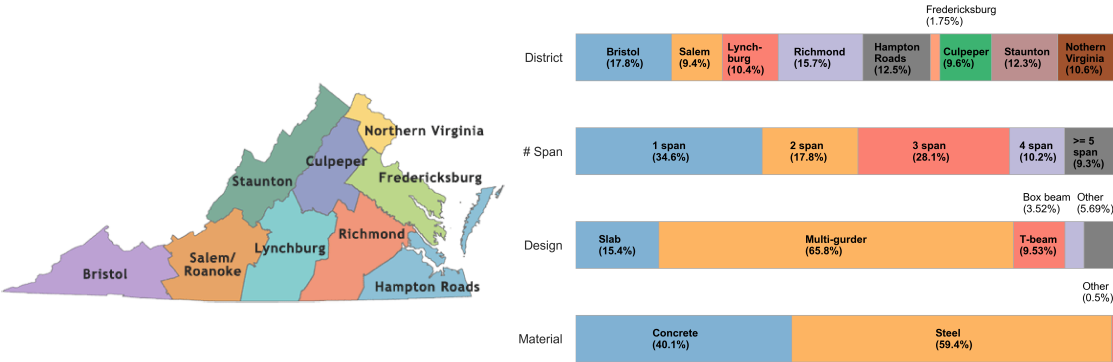


Figure 4.2: Composition of the collected reports in terms of VDOT districts, number of spans, structural designs, and materials.

Although the condition ratings are documented in each inspection report, these ratings can hardly be directly extracted from the reports since they were recorded in various formats such as photo-format cover pages or tables. In this regard, from the pool of all the collected inspection reports, this study used the reports of the National Bridge Inventory (NBI) [1] bridges (bridges that are more than 20 feet long and used for vehicular traffic), whose condition ratings can be obtained from the NBI database. The federal bridge ID and inspection year were extracted from the inspection reports, usually from the first sentence of the report texts, via regular expression matching. The bridge ID and inspection year were then used to match an inspection report with the associated

condition rating in the NBI database.

Images were extracted from the inspection reports using the Zipfile python package [196] that directly accesses the images from the metadata of the reports in the Microsoft Word document format. Irrelevant images such as icons and logos were removed manually while sorting images by sizes to reveal the icons and logos that were mostly smaller than 20 Megabytes; structural drawings were removed manually while sorting by the percentage of white pixels in an image. As images from a report are not readily organized by bridge components of Deck, Superstructure, and Substructure, this study trains an identification tool that separates the images regarding local deficiencies of bridge deck (deck images) from the rest (non-deck images). Figure 4.3 and 4.4 presents example deck and non-deck images from the manually labeled dataset, respectively. The identification tool was developed by fine-tuning a pre-trained DenseNet [118] image classification model using a manually labeled dataset with 1214 images. The deck images were manually identified from 400 reports that were randomly sampled from the collected inspection reports, resulting in 607 deck images. The rest of the images from these reports were non-deck images that illustrate bridge components other than the decks, from which the same number (607) were sampled to form the dataset for developing the identification tool. 90% of the dataset was used for training the identification tool, while the rest 10% were used for hold-out evaluation. Model training used the Adam optimizer [197] with

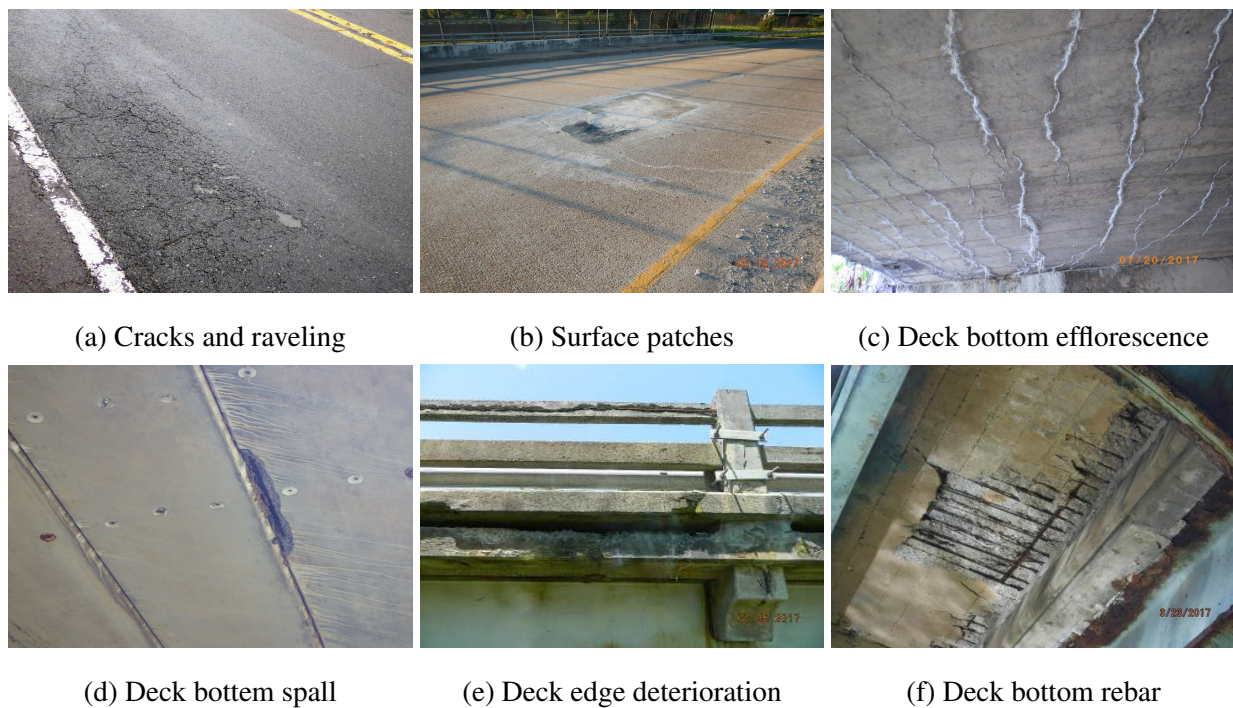


Figure 4.3: Example deck images illustrating local conditions of bridge decks.

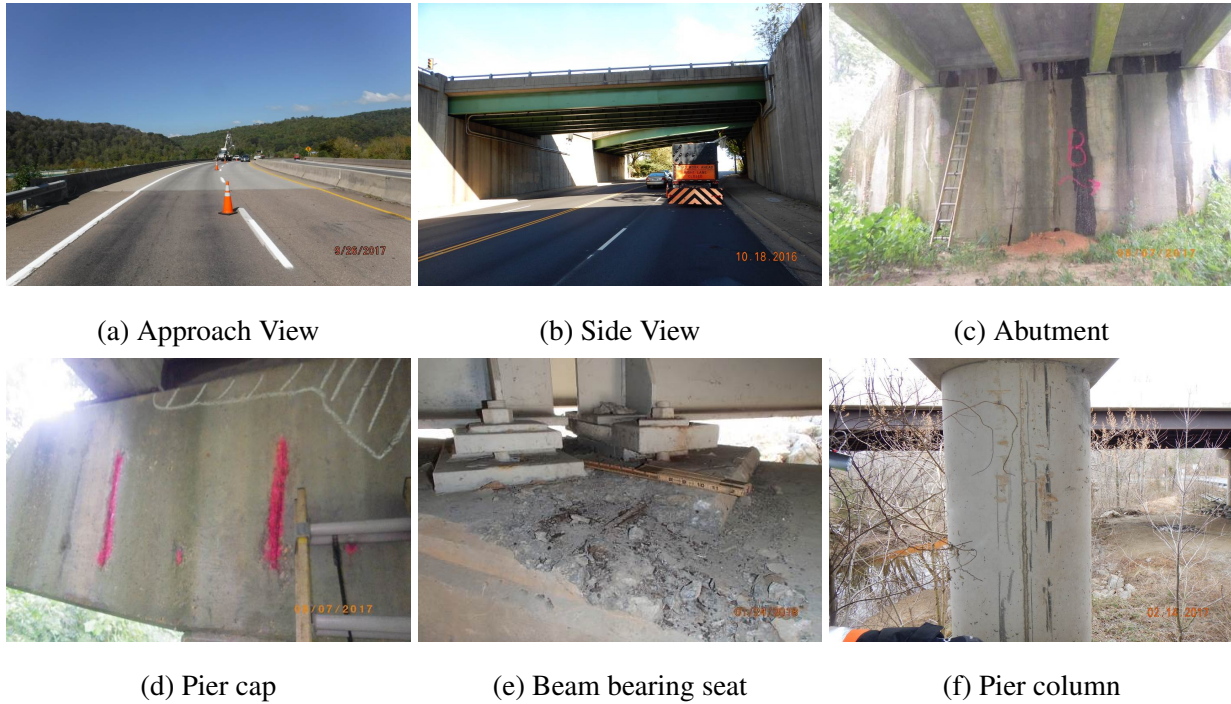


Figure 4.4: Example non-deck images.

a learning rate of $3e-5$ that was selected by searching in the range of $[1e-5, 3e-5, 5e-5, 1e-4]$. This identification tool achieved 88.43% accuracy during testing, and was then used to identify deck images from all the collected inspection reports. The identified deck images from each report were organized in a folder named by a unique index, which is then used to associate the images with the textual data from each inspection report.

Sentences were extracted from the inspection reports using the python-docx package [175] that supports reading table contents from Word document cell by cell and extracting textual content from each cell of the table. Since each inspection report maintains a large table of all its contents, sentences were extracted from the table while keeping the original organization in the table. The sentences describing bridge deck conditions were used in this study to construct the condition rating model. The raw descriptive sentences were tokenized into individual words based on whitespace, standard contractions, and punctuation using the Natural Language Toolkit(NLTK) [177] tokenizer. Tokens were reduced to lower cases and special characters were removed. Measurements in engineering conventions such as $1\text{-}3/4$ were converged to integers; units in punctuation forms such as ' (feet) and ' (inch) were changed to the textual forms. The cleaned tokens were joined by whitespaces and grouped into sentences using the NLTK sentence grouper that was pre-trained on large corpora for identifying the boundaries of sentences. The sentences of cleaned tokens from each

report were organized by the report's index to be combined with the images.

The above data collection and preparation process resulted in a collection of 6670 unique bridges and 6881 reports (211 bridges had more than one corresponding reports from previous inspections). For the 10,521 NBI bridges in Virginia, 3640 bridges were not included in this collection due to the causes including reports in a PDF format; not containing deck images; not listing the inspection year in the beginning; or not organized by bridge components of Deck, Superstructure and Substructure. It is anticipated that a unified representation of the inspection results (images and descriptive sentences) may allow a wider coverage of bridges for constructing automated condition rating models.

4.5 Results and Discussion

The collected reports were randomly split into the training and testing sets with a ratio of 9:1. Table 4.2 presents the statistics of the splits. The training and testing sets were ensured to have a similar distribution of condition categories by stratified random sampling. The reports of the same bridge obtained from different years of inspections were assigned to either training or testing set to achieve full independence between the training and testing sets. This resulted in 6004 bridges (6195 reports) for training and another 666 bridges (686 reports) for testing. The training data was used for model development, while the testing data was held unseen during training for objective evaluation of model performance. It should be noted that bridges with a condition rating of 3 or lower are closed to traffic due to severe structural deficiencies that raise safety concerns; bridges

Table 4.2: Statistics of the training and testing datasets (Top row: training; Bottom row: testing).

Condition	4	5	6	7	8
#Reports	313	991	2,019	2,502	370
	33	110	224	278	41
Avg #Images	5.67	5.20	4.63	3.76	2.44
	4.79	5.86	4.70	3.90	2.78
Avg #Sentences	33.52	27.36	21.76	15.81	7.62
	31.97	36.02	22.38	15.05	8.55

with a condition rating of 9 are newly constructed or renovated without local deficiencies. These condition categories do not represent operational bridge conditions and therefore are not included in the target condition ratings.

4.5.1 Multi-modal Rating Performance

The proposed multi-modal rating model was trained using mini-batch stochastic gradient descent (SGD) optimization, which descends the parameters along their gradients to minimize the loss as defined in Equation 4.15. The batch size were set as 64. A momentum of 0.9 were used, which keeps a portion of previous parameter updates in order to stabilize the training process [178]. Dropout training was applied to the embedding layer that randomly drops the embeddings, which is a regularization technique to alleviate over-fitting [179]. The hyper-parameters of model training including the learning rate, weight decay (L2 regularization), the dimension m of the image vector v_k (Equation 4.2) and the sentence vector s_j (Equation 4.7), as well as the dropout probability, were selected by performing a 5-fold cross validation (CV) grid-search over the following hyper-parameter space discretization: learning rate [$5e-3$, $1e-2$, $3e-2$]; weight decay [0 , $1e-6$, $1e-4$]; vector dimension [100 , 150 , 200]; and dropout probability [0 , 0.3 , 0.5] (selected values were underlined). To enhance the variability of the image data during training, the images were resized to 256×256 and randomly cropped to the input size of the DenseNet (224×224). Other augmentation transforms [117, 118] such as horizontal or vertical flip (with a probability of 0.5), and random rotation (maximum 120 degrees) were also applied to the cropped patches. The random parameters of the crop, flip, and rotation transforms were generated from Uniform distributions. The model were implemented using the PyTorch [181] package that supports GPU-accelerated tensor computations and automated differentiation for optimization in neural networks. The training process was supported by NVIDIA Tesla P100 GPU nodes provided by the University of Virginia’s High-Performance Computing (HPC) servers.

Figure 4.5 presents the training and validation curves of the proposed multi-modal rating model. While the training curve increases through the epochs, the validation ACC0 gradually converges after around 30 epochs. Figure 4.6 presents the confusion matrix of the testing predictions. The confusion matrix demonstrated a clear diagonal concentration where the model rarely mis-predict by more than one rating level. The percentage of “off-by-2” mis-predictions (*e.g.* ground truth condition rating 5 mis-predicted as 7) is 2.19%. Also, the average difference between the predicted

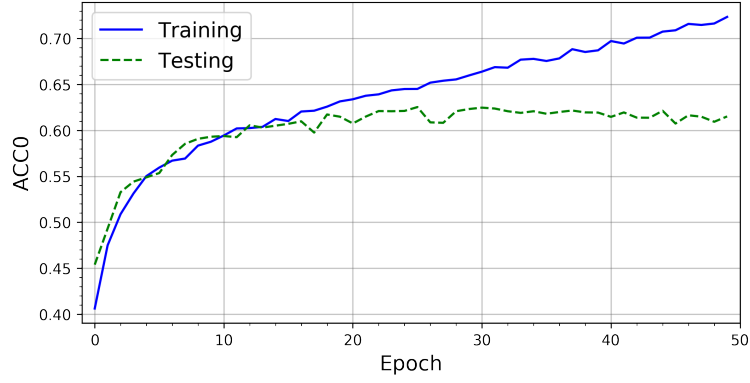


Figure 4.5: Training and validation curves of the proposed multi-modal rating model.

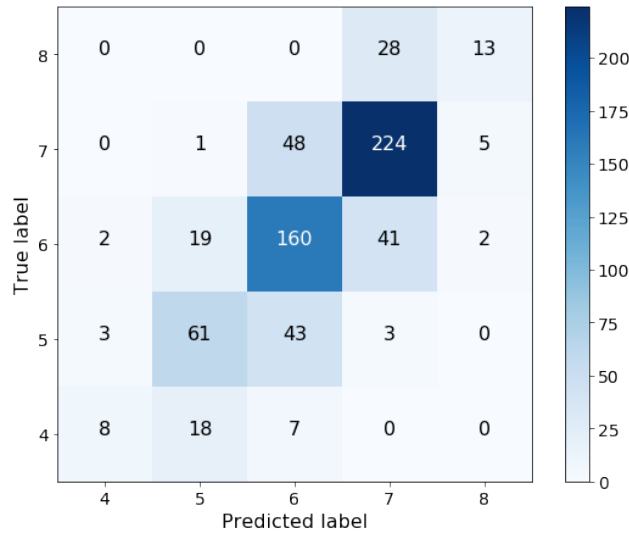


Figure 4.6: Confusion matrix of the testing predictions

ratings and the ground truth is 0.386 and 0.343 as measured by the MSE and MAE metrics, respectively. This performance of the proposed multi-modal rating model is also presented in Table 4.3, with the ACC0 and ACC1 values of 67.93% and 97.81%, respectively.

While the proposed model achieved a relatively small MSE and MAE as well as high ACC0 and ACC1 values during testing, a closer examination of the confusion matrix reveals increasing rating difficulties towards the ends of the condition rating spectrum. A potential cause for this behavior can be the severe imbalance of the sizes of the condition categories. As seen in Table 4.2, over 72% of the inspection reports belong to the condition categories of 6 and 7, and the condition categories of 4 and 8 only take 5% and 6% of the data, respectively. In fact, the U.S. bridge population naturally concentrates on the middle of the condition rating spectrum as the majority of the bridges were constructed over 50 years ago and extensive efforts have been made by the bridge management agencies

to reduce the severely deteriorated bridges in condition 4 and 5 [176]. The lack of regulations in image data collection further added to the imbalance issue. The inspection personnel takes a selective collection of site images, usually of the pressing issues identified during bridge inspections. The minor local deficiencies in the condition 8 bridges were not often documented by images, which further reduced the available multi-modal data at the end of the condition rating spectrum. The class imbalance is known to affect the classification performance of especially the minority classes [182, 184–186]. Two major strategies to combat class imbalance include re-weighting the cost of mis-classification [187, 188] and re-sampling the training data [189] (over-sampling the minority class or under-sampling the majority class). The use of the re-weighting (median class frequency balancing [190]) and re-sampling strategies did not yield improvement in the distribution of accuracy, which may be attributed to the fact that the minority condition categories only contain around 300 data that are limited in revealing the underlying trends even with balancing strategies. Future work may collect inspection reports from multiple states to populate the minority classes, which may allow for effective re-weighting or re-sampling strategies.

The proposed model is compared with two uni-modal baselines (image-based and text-based) and two multi-modal variations that perform fusion at the score level and the document level, respectively.

Image The Image baseline extracts image vectors the same way as the proposed fusion model (Equation 4.1, 4.2), and combines the extracted image vectors for condition rating. The hyper-parameters of the learning rate, weight decay, the dimension m of the image vectors $v_k, k \in [K]$, and the method for combining the image vectors were selected by performing a 5-fold cross validation (CV) grid-search over the following hyper-parameter space discretization: learning rate [5e–3, 1e–2, 5e–2]; weight decay [0, 1e–6, 1e–4]; vector dimension [100, 150, 200]; combining method [averaging, max pooling] (selected values were underlined).

Text The Text baseline adopts the HAN model [164] that constructs the sentence vectors the same way as the proposed model (Equation 4.7), and has another encoder-attention module to combine the sentence vectors into a document-level representation for condition rating. The hyper-parameters for the Text baseline were selected from the same hyper-parameter space discretization as the proposed model, resulting in selecting a learning rate of 0.01 and a vector dimension of 100 with a weight decay of 1e–6.

Score The Score fusion model takes the softmax scores (as computed by Equation 4.14) from the two uni-modal baselines (Image and Text), and calculates a weighted sum of the scores to fuse the two modalities. The weights of the Image and Text scores, denoted as w_i and w_t , were selected as 0.1 and 0.9, respectively, by searching from $w_i = [0.1, 0.2, \dots, 0.9]$, $w_t = 1 - w_i$.

Doc While the proposed fusion model directly combines the image and sentence vectors for fusion, the Doc fusion model first combines the image vectors and the sentence vectors separately, using the same method as in the uni-modal baselines, and then concatenate the combined visual and textual representations for fusion. The hyper-parameters were selected from the same hyper-parameter space discretization as the proposed model, resulting in selecting a learning rate of 0.005 and a vector dimension of 100 with a weight decay of $1e-6$.

Table 4.3 presents the performance of the proposed fusion model compared with the uni-modal baselines and multi-modal variations. The image modality alone achieved 49.42% ACC0 in classifying condition ratings of four through eight. Although the majority of existing defect recognition models are image-based, the image data from bridge inspection reports alone does demonstrate sufficiency in supporting automatic condition rating using the image baseline model. The Text baseline significantly improved the condition rating performance in all metrics compared with the image baseline, which confirmed the hypothesis that the textual data from bridge inspection reports contain more comprehensive information regarding bridge conditions as compared with the visual data, and the visual modality is supplemental modality compared with the textual modality. By di-

Table 4.3: Performance of the proposed fusion model compared with the uni-modal baselines and multi-modal variations.

Model		ACC0 (%)	ACC1 (%)	MSE	MAE
Uni-modal	Image	49.42	91.11	0.802	0.601
	Text	64.43	98.10	0.413	0.375
Fusion	Score	65.16	97.96	0.410	0.369
	Doc	67.20	98.25	0.380	0.345
	Proposed	67.93	97.81	0.386	0.343

rectly fusing the softmax scores from the Image and Text baselines via the Score fusion, the fused ACC0 demonstrated minor improvement of 0.73%, while the ACC1 slightly reduced by 0.14%. The weight assigned to the image and text scores were 0.1 and 0.9, respectively, indicating that the Score fusion model only fused in a small portion of the image information. Further increasing the weight on image scores decreased the ACC0. The minor improvement of the Score fusion over the text baseline showed that the direct Score fusion method adds limited benefits in fusing supplemental modality like the visual data in bridge inspection reports. Both the Doc fusion and the proposed fusion models both significantly improved over the Text baseline in terms of ACC0, and the mean error metrics. The difference between the Doc fusion and the proposed fusion is whether to summarize within modality first before combining the multi-modal representations. With similar modules (in different orders) and the number of parameters, the Doc fusion and the proposed fusion models achieved similar performance. Allowing the image vectors and sentence vectors to be jointly encoded by the fusion encoder and attention mechanism as in the proposed model slightly outperformed the Doc fusion in terms of ACC0. It should be noted that the Score and proposed fusion did not outperform the Text baseline in terms of the ACC1 metric, which might be attribute to the fact that all fusion models and the Text baseline achieved relatively high ACC1 (98%) with only small fluctuations in the scores.

Impact of model components The proposed model adopts a fusion module that contains components including layer normalization, sequence encoder, and the attention mechanism. To examine the impact of each component, Table 4.4 presents the performance of the proposed model while removing each component. The attention mechanism is replaced with averaging or maximization to combine the encoded representations of images and sentences. As revealed by the ACC0, MSE, and MAE metrics, removing each component in the proposed model caused reductions in the rating performance. Base on the reduction caused by removing each component, the sequence encoder is the most important component in the model architecture, followed by the attention mechanism and the Layer Norm. The removal of each model component caused fluctuate performance in the ACC1 metric instead of the trend as revealed by the other three metrics, removing the layer normalization and replacing the attention mechanism with maximization did not reduce the rating performance compared with the proposed model. This might be attributed to the fact that all the model variants achieved similar performance as evaluated by ACC1. Considering that one mis-prediction in the testing set causes 0.15% reduction in ACC1, the fluctuations in ACC1 are not as strong support for

the analysis of performance changes as compared with the other three metrics.

Table 4.4: Impact of the components in the proposed architecture.

Model	ACC0 (%)	ACC1 (%)	MSE	MAE
Proposed	67.93	97.81	0.386	0.343
w/o LN	-2.04	0.29	0.012	0.017
w/o Encoder	-4.08	-0.14	0.053	0.043
w/o Attn (Max)	-2.62	0.00	0.027	0.026
w/o Attn (Ave)	-7.58	-1.16	0.111	0.087

Impact of visual feature extractors To examine the impact of different visual feature extractors on the multi-model rating model, Table 4.5 presents the performance of the proposed model with different architectures used for visual feature extraction. While The proposed model adopts DenseNet (161 layers) to extract vector representations from the images, two variants of the well-known Resnet were also examined, the medium size Resnet50 with 50 layers, and the large size Resnet152 with 152 layers. The Resnet introduces residual connections to the deep convolutional neural network to improve optimization, and the DenseNet further adds dense connections among layers to encourage feature reuse and strengthen feature propagation. Regarding the hyperparameter setting, both Resnet variants were trained using the same vector dimension and dropout probability as the proposed model, while the learning rate and weight decay were tuned for each Resnet variant, resulting in a learning rate of 0.01 with no weight decay for Resnet50 and a learning rate of 0.03 and a weight decay of $1e-6$ for Resnet151. As shown in the table, the Resnet50 and the Resnet152 obtained similar condition rating performance, while the Resnet50 slightly outperformed the Resnet152. Both Resnet variants did not reach the performance of the proposed model with Densenet161 as visual feature extractor, which demonstrated the strength of the dense connections in the pre-trained DenseNet architecture in extracting informative visual features. All three models demonstrated improvements over the uni-modal baselines as presented in Table 4.3, indicating the strength of fusion despite different visual feature extractors.

Table 4.5: Impact of visual feature extractors.

Model	ACC0 (%)	ACC1 (%)	MSE	MAE
Resnet50	66.91	97.81	0.397	0.353
Resnet152	66.47	97.38	0.414	0.362
DenseNet161	67.93	97.81	0.386	0.343

Impact of dropout To examine the effect of dropout training, Figure 4.7 depicts the training and validation curves for the proposed multi-modal rating model with different dropout probabilities. As shown in the figure, dropout training (with a probability of either 0.3 or 0.5) alleviated over-fitting, which obtained smaller gap between the training and validation curves compared with no dropout training. Besides, the validation ACC0 was able to converge to a higher value when increasing the dropout probability from 0.3 to 0.5, which demonstrates the benefit of dropout training for improved learning and reduced over-fitting.

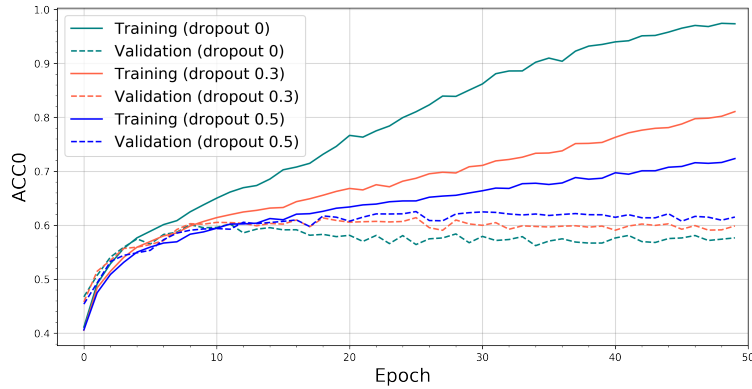


Figure 4.7: Training and validation curves with different dropout probabilities.

4.5.2 Uncertainty Analysis

The training of the proposed model introduced variability via data augmentation and dropout in order to alleviate over-fitting and promote the learning of robust mappings that are generalizable to unseen testing data. As discussed in Section 4.5.1, the images were augmented via random crop, flip (horizontal and vertical), and rotation, where the corresponding random parameters were sampled from uniform distributions. Dropout training was applied to the embedding layer with a probability

of 0.5 that randomly set the embedding of the input words to zero. To quantify the uncertainty introduced by such training strategies, this section applies data augmentation and dropout during testing and performs repeated testing experiments to analyze the distribution of model predictions. The mean of this distribution is used as the predicted rating, while the standard deviation of this distribution is used as the measure of model uncertainty. As the data augmentation can be considered as having input images from variant directions and angles, and the dropout training over the embedding layer is equivalent to randomly omitting words in the input sentences, the uncertainty obtained from this analysis can be interpreted as model uncertainty given these variations in the input images and sentences.

The testing experiments were repeated for $\mathcal{T} = 100$. The maximum uncertainty of the testing predictions is 0.817, while 95% of the testing predictions resulted in uncertainty within 0.535. It should be noted that, based on the reliability study conducted by the Federal Highway Administration (FHWA), where a sample of bridge inspectors across the states was tasked to assigned condition ratings for the same bridges, it was estimated that 95% percent of the condition rating for the entire bridge population vary within approximately two rating points from average. The reliability study can be considered as an estimation of rating uncertainty introduced by multiple factors including inspector variance, state variance, and environment variance (*e.g.* lighting, traffic, etc.). The proposed model learns from the aggregated knowledge base of how bridge condition has been assessed by the inspectors across the states, and mapped the multi-modal data from inspection reports to the condition ratings. While the computed uncertainty of predictions is only introduced by disturbance of data augmentation and dropout, this uncertainty is significantly smaller compared with the estimated variance from the reliability study, which demonstrated that learning from large aggregated historical condition ratings can be a viable approach in alleviating the inevitable inconsistency issue associated with the human condition rating process.

Figure 4.8 presents the histogram of model uncertainty given correct or incorrect predictions. As seen from the figure, the correct predictions distribute more over smaller uncertainty values (below 0.3) compared with incorrect predictions. Accordingly, the incorrect predictions distribute more over larger uncertainty values (above 0.3). Motivated by such property of the model uncertainty, Figure 4.9 illustrates that incorporating the consideration of uncertainty can practically improve the condition rating performance. As illustrated in 4.9, the rating accuracy increases as the threshold of allowable uncertainty decreases (more strict with uncertainty). The ACC0 reaches 77.16% when

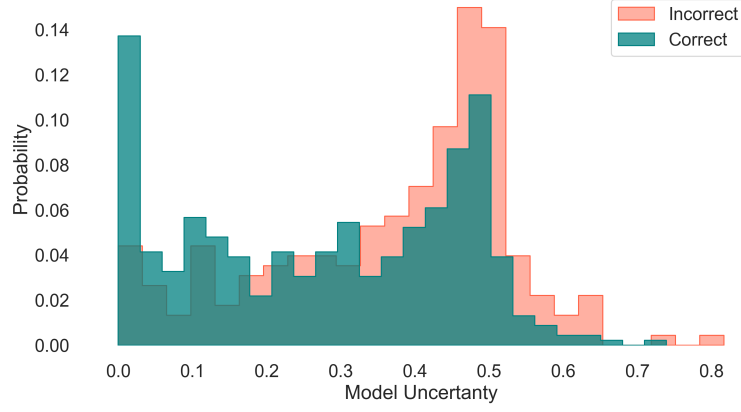


Figure 4.8: Histogram of model uncertainty given correct/incorrect predictions.

the maximum allowed uncertainty is 0.327, which corresponds to 47.5% of the testing data. When applying the condition rating model with the consideration of model uncertainty, the most uncertain rating data points may be referred to human inspectors for further investigation, and the remaining more certain predictions can achieve improved performance.

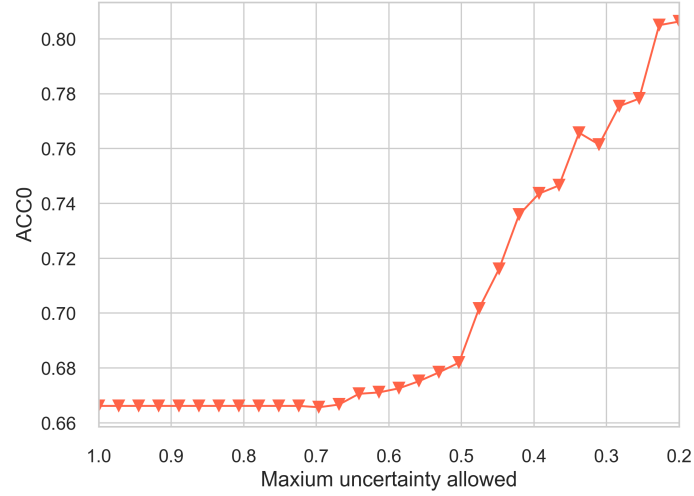


Figure 4.9: Accuracy gain while considering model uncertainty.

Figure 4.10 presents the model uncertainty in different condition rating categories. The mean uncertainty decreases as the condition category increases from 4 to 7, which indicates an increasing difficulty in rating bridges in poorer conditions. The mean uncertainty of condition 8 was higher than condition 7, which may be attributed to the fact that a limited number of condition 8 data is available in this study.

The uncertainty evaluation disturbed the model with image augmentation transforms and dropout, and generated $\mathcal{T} = 100$ predictions for each testing data point. Each prediction corresponds to 5

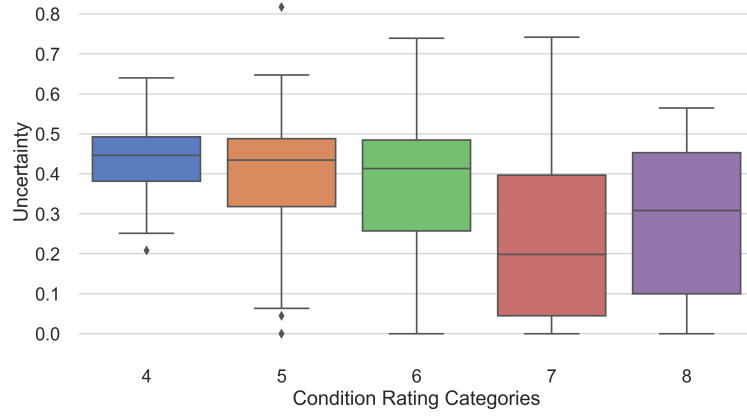


Figure 4.10: Model prediction uncertainty in different condition categories.

softmax scores, one for each condition category, as computed in Equation 4.14. To investigate the rating behavior among different condition categories, the correlation matrix of the 5×100 softmax scores was computed and Figure 4.11 presents the average correlation over the testing data, where the green cells illustrate positive correlation, and the yellow and red cells illustrate negative correlation. The positive correlations demonstrated difficulty in distinguishing the neighboring condition categories, *e.g.* the condition 5 scores are correlated with the scores of condition 4 (0.34) and 6 (0.3). Along with the same trend, the condition 8 scores are correlated with the condition 7 scores (0.39). In contrast, the negative correlations indicate the rating model’s capability in distinguishing between non-neighboring conditions. For example, the condition 5 scores have a strong negative

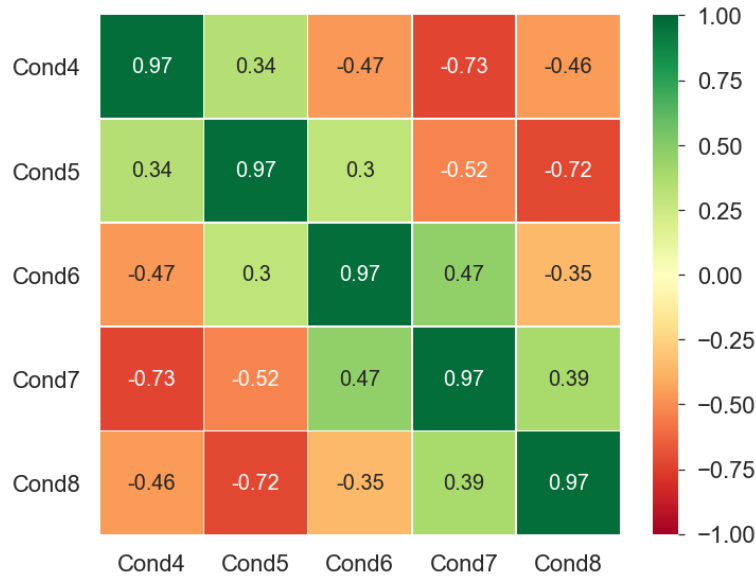


Figure 4.11: Correlation among predicted softmax scores of different condition categories.

correlation with conditions 7 (-0.52) and 8 (-0.72), which shows that while considering the uncertainty of model predictions, having high softmax scores in condition 5 leads to low scores in condition 7 and 8. In other words, the model is not likely to mis-predict outside the neighboring condition categories of the ground truth.

Chapter 5

Context-aware Sequence Labeling for Condition Information Extraction from Historical Bridge Inspection Reports

Li, T., M. Alipour, and D.K. Harris, Context-aware sequence labeling for condition information extraction from historical bridge inspection reports. *Advanced Engineering Informatics* (in review), 2020.

5.1 Abstract

Effective upkeep of aging infrastructure systems with limited funding and resources calls for efficient bridge management systems. Although data-driven models have been extensively studied in the last decade for extracting knowledge from past experience to guide future maintenance decision making, their performance and usefulness have been limited by the level of detail and accuracy of currently available bridge condition databases. This paper leverages an untapped resource for bridge condition data and proposes a new method to extract condition information from it at a high level of detail. To that end, a natural language processing approach was developed to formalize structural condition knowledge by formulating as a sequence labeling task and modeling inspection narratives as a combination of words representing defects, their severity and location, while accounting for the context of each word. The proposed framework employs a deep-learning-based approach and incorporates context-aware components including a bi-directional Long Short Term Memory

(LSTM) neural network architecture and a Conditional Random Field (CRF) classifier to account for the context of words when assigning labels. A dependency-based word embedding model was also used to represent the raw text while incorporating both semantic and contextual information. The sequence labeling model was trained using bridge inspection reports collected from the Virginia Department of Transportation bridge inspection database and achieved an F1 score of 94.12% during testing. The proposed model also demonstrated improvements compared with baseline sequence labeling models, and was further used to demonstrate the capability of detecting condition changes with respect to previous inspection records. Results of this study show that the proposed method can be used to extract and create a condition information database that can further assist in developing data-driven bridge management and condition forecasting models, as well as automated bridge inspection systems.

5.2 Introduction

Bridge infrastructure systems face significant preservation challenges characterized by aging and deterioration as well as the shortage of funding required for maintenance, rehabilitation and repair (MR&R) operations. Analysis of the 2019 National Bridge Inventory (NBI) database reveals that nearly forty percent of the bridges in NBI are 50 years or older, 8.4% are designated structurally deficient, and over 11.3% have posted weight limits restricting the flow of traffic. Federal Highway Administration (FHWA) estimates that the backlog of national bridges' rehabilitation projects reaches as high as \$123.1 billion. An annual investment of \$24.6 billion is needed to clear this backlog in the twenty year period (2012-2032), while in 2012, only \$16.4 billion were spent on the maintenance of all bridges [20].

Effective upkeep of such an aging and deteriorating bridge network in the face of limited funding and resources calls for the integration of efficient automation concepts into bridge management systems. In this regard, data-driven methods have recently attracted significant attention in the research community. The success of these approaches relies extensively on the existence of accurate and detailed databases of bridge conditions. The methods proposed in this paper are envisioned to bridge the information gap caused by the existing global and coarse condition characterizations through the extraction and incorporation of fine-grained local condition information in addition to component-level and element-level ratings. This, in return, will enable the bridge management agencies to identify and respond to the maintenance needs in a more accurate and timely fashion,

thus contributing to improvements in the state of infrastructure.

Considering the importance of condition data in infrastructure monitoring, deterioration modeling and forecasting, and maintenance decision making, the research community has focused on different sources and methods of automated condition information extraction. As visual inspection remains the predominant approach for assessing the condition of bridge components and elements [198], recent literature in this field shows a significant number of publications on image-based inspection automation methods. These works aim to extract structural condition information from image data obtained using a variety of ways ranging from robotic inspection [29, 30, 33] to crowd-sourced monitoring [199–201], and different levels of detail ranging from images and image patches [47, 48] to pixel-level defect detection [44–46]. While structural imagery presents a powerful resource for condition assessment and subsequent maintenance decision-making, this paper identifies and focuses on *textual* data as another rich source of condition data that can be integrated into automated infrastructure inspection and management systems. Such textual data can contain a high level of detail describing the structure and its components, the types of detected defects, and their location, severity (qualitative), and size measurements (quantitative) as described in the form of natural language narratives by observers or professional inspectors.

To access and leverage this high level of detail, this paper introduces an untapped resource for bridge condition data and an innovative method to extract condition information. Bridge inspection reports have been maintained by state Departments of Transportation (DOT) through years, documenting first-hand condition information, and can greatly benefit bridge management decision making if fully exploited. The National Bridge Inventory Standard (NBIS) requires routine inspection for most bridges and culverts biennially [1]. With each inspection, bridge inspection reports are filled out by the inspectors according to the Bridge Inspector’s Reference Manual [2]. These reports are rich in details that record and document local deficiencies at a specific point in time, and are used by inspectors to track the progression of defects compared with the previous inspections. This valuable information is, however, buried in the inspection reports in the raw natural language format and cannot be directly used to support bridge deterioration analytics. Existing rule-based Natural Language Processing (NLP) applications in civil infrastructure management have contributed to the successful information extraction from other textual data in the domain. However, considering the heterogeneous nature of the inspection reports composed by various professional inspectors, it can be difficult to define a set of all-encompassing rules to cover variant scenarios. Existing machine-

learning-based (ML-based) applications usually leverage syntactical features [39, 40] or word frequency features [38, 41, 42], which limits the model’s ability to exploit the semantics contained in the inspection reports texts. While the use of context-aware deep-learning-based methods is scarce in the field of infrastructure management, the complexity of inspection report texts demands such a model to capture the various use of words and extract accurate information.

Figure 5.1 shows a framework for infrastructure maintenance information retrieval from bridge inspection reports. This framework aims to extract context-aware condition information, standardize it based on standard element and defect definitions, and uses it for bridge maintenance analytics. This paper focuses on the context-aware condition information extraction task by developing a natural language processing approach that analyzes the textual descriptions from bridge inspection reports and extracts condition-related chunks. This information extraction task is formulated as a sequence labeling scheme that analyzes each sentence and assigns a condition category label to each word based on its context to indicate the typical information of interest when evaluating the condition of a bridge such as the local deficiency (type), its measurements (severity), and where it is located (location). The sequence labeling model integrates context-aware components to enforce the consideration of contexts when assigning the label for each word. The assigned labels naturally group the sentence into chunks belonging to different categories. The extracted chunks are then

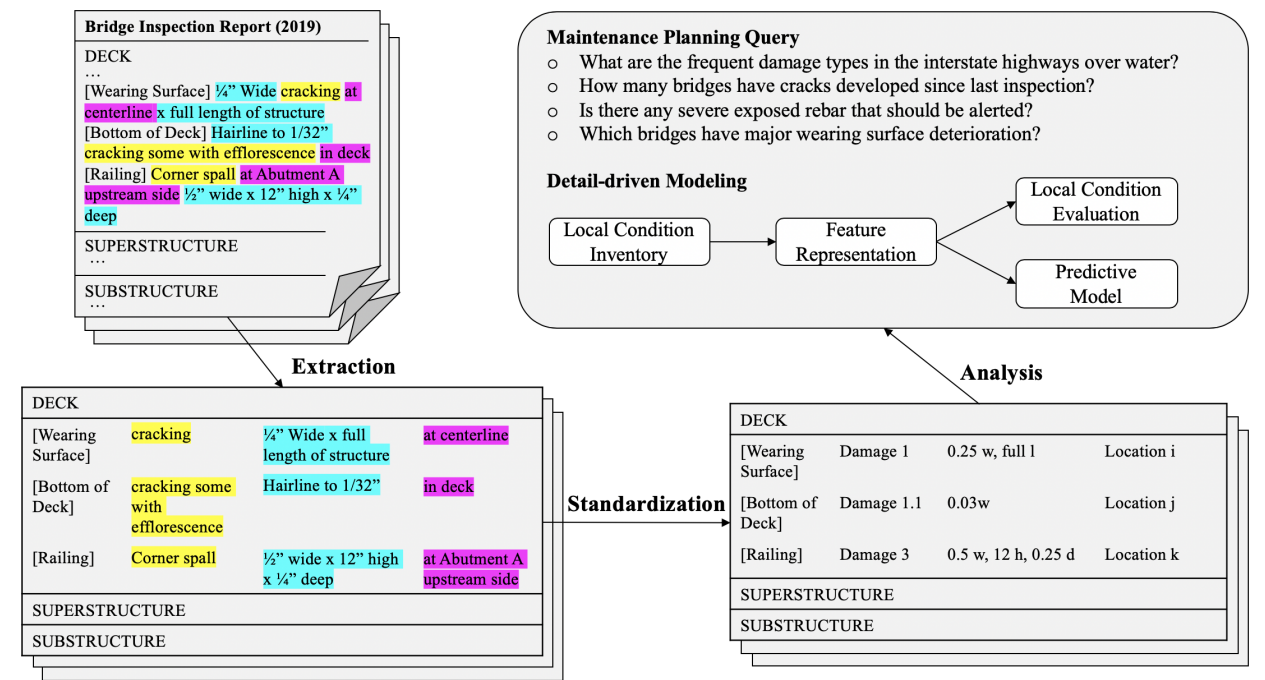


Figure 5.1: Framework for infrastructure maintenance information retrieval.

standardized according to standard bridge element and defect definitions (e.g. those in AASHTO's Manual for Bridge Element Inspection [16]) to form a condition inventory that can provide features for deterioration prediction and other system-level bridge data analysis, as presented in Figure 5.1. Such a condition inventory can be used to support future research efforts in data-driven bridge condition modeling and maintenance decision making. Detailed bridge deterioration models can also be constructed based on the information extracted from the proposed model with its rich, historical, and localized condition details.

5.3 Research Motivation

5.3.1 Identified Limitations and Knowledge Gaps

Among the reviewed literature, the proposed model is most related to [39] in the extraction of information from bridge inspection reports by assigning sequential labels. In their work, a semi-supervised Conditional Random Fields model was constructed based on a set of syntactical and ontology look-up features. Eleven labels were defined including bridge element, deficiency, cause, material, numerical measurements, as well as an 'Other' category for words that do not belong to any of the designed categories. The model was trained on the 2006 I-35W Mississippi River Bridge inspection report and was tested on 11 other bridge inspection reports. While their model achieved a relatively high level of labeling accuracy (90.7% F1 score), a number of limitations exist in the following aspects. First, the dependency on ontology features necessitates the creation of an ontology for each specific domain of interest. The labels were designed based solely on their meanings and regardless of their roles with respect to bridge conditions, and thus do not link directly to bridge condition interpretation such as the location and severity of a certain damage condition. Additionally, the labeling model only considers a context window of size one, *i.e.* each word is assigned a label based on features of the words right before and after it, and did not fully exploit the long range dependencies between words. Finally, more than half of the words fell into the 'Other' category which might lead to the loss of condition information such as negation or positional information. Conversely, this study develops a context-aware sequence labeling model, where the labels help group each sentence into chunks of information that are typically of interest to maintenance decisions such as type, location and severity. Extracting chunks of information instead of individual words is especially important for bridge inspection. For example, instead of extracting

the word ‘2’ as number followed by the word ‘feet’ as unit, it is important to know when ‘2 feet’ is used to describe location of a spall and when it is used to describe the size of a spall, since the two situation raise different levels of concerns in bridge inspection. In the sequence labeling task to extract chunks of information, the labels should be assigned based on their context such that the same word can be assigned different labels when in different contexts that refer to different categories of condition information. It is therefore hypothesized that accounting for the dependencies between the neighboring words that build up the context of each word can improve the quality of the assigned labels. To that end, this study formalizes structural condition as the central topic of inspection and maintenance by modeling inspection narratives as a combination of defect names (N), their location (L), and severity (S) arranged in the heterogeneous and complex patterns of natural language to describe an infinite variety of real-life inspection scenarios. Multiple mechanisms were designed and implemented to enforce these dependencies and contextual relationships, resulting in a context-aware condition mapping. The context-awareness mechanisms employed herein include the use of dependency-based word embeddings, bidirectional Long Short-Term Memory (LSTM) architecture, and the use of a Conditional Random Field (CRF) model. Specifically, this paper goes beyond the existing literature in this domain by increasing the length of the context window from one to include longer-range dependencies among the words while performing the sequence labeling task.

This paper aims to be a part of the growing body of literature that focuses on formalizing engineering knowledge through the use of automated text mining from digital engineering documents (e.g., patents [202, 203], accident reports [109], design documentation [204, 205]). Given the ever-increasing proliferation of such data in the domain of construction and infrastructure, this field of research has been identified as a vibrant area that is positioned to affect the current discourse in the field of advanced engineering informatics (AEI) [206]. While this paper belongs to this body of work, it aims to bridge the gap that exists due to the lack of studies on bridge inspection reports as the source of information extraction. In other words, this research identifies textual bridge inspection narratives as an untapped resource for bridge condition information. In the absence of formal knowledge representation such as the framework proposed in this paper, such raw data has not been used to support bridge management analytics.

5.3.2 Need for Context-aware Information Extraction

A fundamental requirement for an intelligent information extraction system is its ability to consider semantics and the context of a word in its predictions. In other words, the desired sequence labeling model should not memorize every word and assign a label in a dictionary-lookup fashion, but should predict the label based on the meaning of the word and its context in the sentence. This is not only because building an all-encompassing dictionary to look up labels is obviously cumbersome, but the benefits of a context-aware labeling system also lie in the flexibility of allowing the same word to be labeled differently in different contexts. The desired labels help group the words in a sentence into chunks of conditions (*e.g.* name, location or severity), instead of trying to classify each word independently into an entity category. This arrangement is rooted in both the way that bridge inspection information is documented, and the desired use case of the extracted information from the inspection reports. Typical information of interest when evaluating the condition of a bridge include the local deficiency (type), its measurements (severity), and where it is located (location), and each piece of information functions only when put into the proper context. The Bridge Inspector’s Reference Manual [2] requires that when documenting a deficiency encountered during inspection, the exact location, severity and extent of deficiencies should be specified to determine the bridge condition. A number of examples provided in this manual that guide the documentation of deficiency quantification and location are shown in Table 5.1.

Table 5.1: Example condition documentation from Bridge Inspector’s Reference Manual [2].

Category	Example Description
Deficiency	2 feet × 3 feet × 2 inches deep
Quantification	4 feet high by full abutment width
	1 foot × 6 inches
Deficiency	Left side of web, top half, 3 feet from north bearing
	Top of top flange, from 3 feet to 6 feet west of Pier 2
Location	7 feet -3 inches from fixed bearing on beam 3 at abutment 1
	3 feet -1inch from west corner of abutment 2

As seen in these examples, a typical practice when documenting bridge conditions in the inspection reports is using the bridge element as reference for deficiency size or location. Therefore,

words representing measurement units (e.g. feet) and bridge members (e.g. abutment) are two examples where the same word should belong to different condition categories in different contexts (see words in bold in Table 5.1). Additionally, the information comes in chunks instead of single words when quantifying or localizing bridge deficiency conditions. To that end, the same word should be allowed to have different labels in different contexts, and the desired labels should naturally segment the sentences into pieces of condition information. This desired feature requires a sequence labeling model that can recognize the semantics and dependencies within the context among the input sequence of words. To that end, this study used a context-aware sequence labeling model that enforces the consideration of context in the following aspects:

1. the dependency-based word embeddings that account for the context of words in their embedding representations;
2. the Long Short-Term Memory (LSTM) cells in the Recurrent Neural Network (RNN) model that emphasize long-term dependencies between words;
3. the bi-directionality of the network that exploits information from the words both before and after the word of interest;
4. the Conditional Random Fields classifier that infers each label based on its neighboring labels.

The details of these model components are discussed in the following sections.

5.4 Research Approach

This study builds an NLP-based information extraction system to characterize bridge condition information from the textual data in bridge inspection reports. The goal of this system was to dissect each sentence in an inspection report into segments of predefined categories (*e.g.* condition name, location, and severity). This problem is formulated as a sequence labeling task that reads each sentence and assigns a category label to each word. The category labels include condition Name (N), Severity (S), Location (L) and Other (O), where the 'Other' category contains words connecting the name, location or severity chunks that are not relevant to bridge conditions. To further categorize the severity descriptions, qualitative severity descriptions and numerical severity descriptions were labeled as two separate categories, S(Q) and S(N), respectively. Figure 5.2 presents the proposed processing pipeline using an example sentence from bridge inspection reports. The original

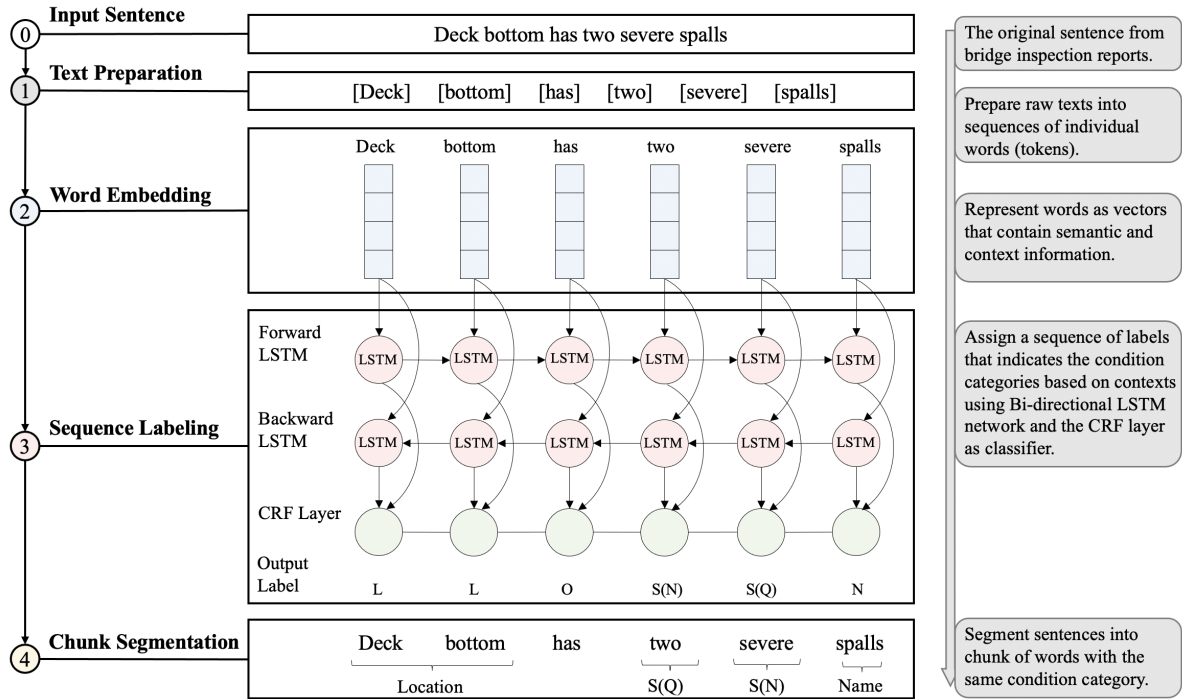


Figure 5.2: The proposed processing pipeline with an example sentence from bridge inspection reports. (N: Name, L: Location, S(Q): Qualitative Severity, S(N): Numerical Severity, O: Other)

input sentence is prepared into a sequence of individual words (tokens) and is then represented by word embeddings that contain semantics and context information. The word embeddings are then fed through the sequence labeling model that consists of Bi-directional Long-Short Term Memory (LSTM) layers and the Conditional Random Field (CRF) classifier. To assign a label for each input token, the bi-directional LSTM layers help encode information before and after the token into a hidden representation, and the CRF classifier further enhances the consideration of neighboring words while assigning a label for the token. The Bi-LSTM-CRF sequence labeling model assigns a label for each token in the input sentence that indicates the condition category of the token. The obtained labels are then used to chunk the input sentence into segments of condition information. The subsections will introduce the details of word embedding and each context-aware component in the sequence labeling model.

5.4.1 Word Embedding

Raw text from the inspection reports needs to be represented by numerical vectors before it could be passed through the sequence labeling model. Traditional representations include one-hot vectors (one element at the index of the word in the vocabulary is set to 1 and all the other elements are

set to 0) and engineered word features such as capitalization, prefix/postfix, or other manually-assigned features. The one-hot vectors suffer from sparsity issues with a dimension as large as the entire vocabulary. The engineered word features require domain- or task-specific efforts. Another representation option, which demonstrated strong performance especially with deep learning-based language models [69], is the dense word embeddings. With dense embeddings, each word is represented by a vector of the same dimension, typically of size 50-300 [207], which is much smaller than the vocabulary size. The word embeddings are initialized and optimized during the training of the deep learning model, which allows the embeddings to learn the internal relationships among the training corpus. The most widely adopted way of obtaining such word embeddings is via a skip-gram model [70], which is developed to encode semantic information into the embeddings. The intuition behind a skip-gram model is that, a word's meaning is given by its context, therefore neighboring words that share the same context should be represented by similar word vectors.

To that end, the skip-gram model trains a simple three-layered neural network that predicts the target context word based on the current word. Figure 5.3 presents the architecture of a skip-gram model. The model uses the input word w_t to predict its neighboring words w_{t-2} , w_{t-1} , and w_{t+1} . Consider the input-neighboring word pair (w_t, w_{t-1}) , denote their indices in the vocabulary as i and j , respectively. The input layer takes in the input word w_t , which is represented by a V -dimensional

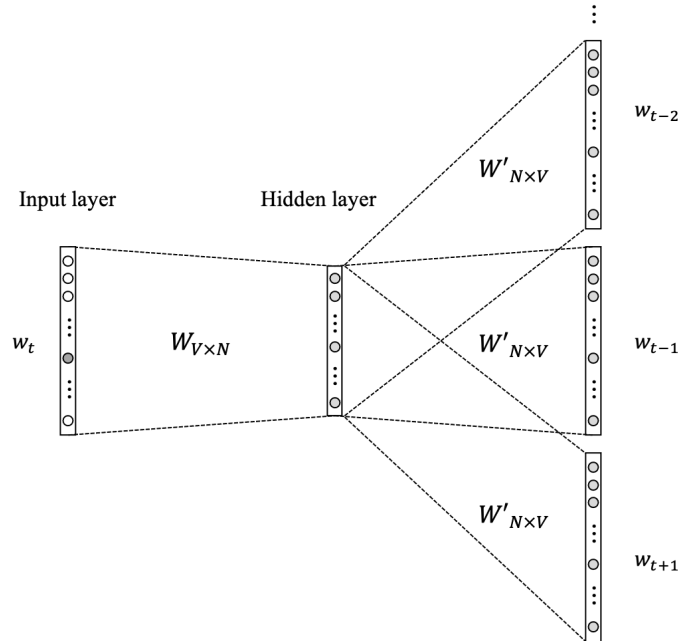


Figure 5.3: Skip-gram model architecture redrawn based on [4]. (w_t : input word; w_{t-2} , w_{t-1} , w_{t+1} : neighboring words; V : vocabulary size; N : word embedding dimension.)

one-hot vector that has 1 for the i_{th} dimension and 0 otherwise (V is the size of the vocabulary). The output layer is a V -dimensional vector that contains softmax scores (probability) of the neighboring word. $W_{V \times N}$ and $W'_{N \times V}$ are the weight matrices that will be optimized during the training process, where N denotes the desired size of dimension for the word embeddings. The hidden layer has no activation functions (operations within each node of a neural network) but simply copies the i_{th} row of $W_{V \times N}$. Considering all of the neighboring words of the same input word, the training process will force the corresponding row vector of the input word to capture the contextual information in order to support the prediction of all its neighboring words. Therefore the row vectors in the weight matrix $W_{V \times N}$ are then rendered as word embeddings. The word embeddings obtained this way are proven to contain semantic information such that word with similar meanings are closer in the embedding vector space [70]. Textual data from the inspection reports were first fed through the word embeddings before being processed by the sequence labeling model. The output from the word embedding process provided a representation that contains semantic and context information of the input textual data, which can be further exploited by the sequence labeling model.

5.4.2 Bi-directional LSTM

In addition to the connections in simple feed-forward neural networks, Recurrent Neural Network (RNN) introduces recurrent connections between the RNN nodes in hidden layers to model the dependency between sequential inputs. To ensure that each label is assigned with the consideration of the longer-range context of the word, Long-Short-Term Memory (LSTM) cells were used in the hidden layer of the RNN. LSTM cells are a type of RNN node that have gate structures especially designed to manage long-term and short-term dependencies. An LSTM cell is designed to contain different operations to organize the long-range context among the input sequence. The operations are carried out through the input gate i_t , output gate o_t , forget gate f_t and the memory cell c_t [208]. Each LSTM block takes x_t as input and produces an output h_t at time step t , where x_t denotes the embedding vector of the input word w_t , which is a corresponding row vector from the embedding matrix $W_{V \times N}$ that was pre-trained using a skip-gram model as presented in Section 5.4.1. The LSTM block carefully organizes the information through the gate and cell operations, which prevents the forgetting of important long-term knowledge. While a regular RNN processes an input sequence one by one and computes the hidden state h_t of the current input x_t based on the hidden state of the previous input h_{t-1} , which only makes use of the previous context for a given word, to label

a word based on its context, obviously both previous and future context should be exploited. To capture this bi-directional information (past and future words in a word sequence), each sentence is passed through the network twice, once forward and once backward. Outputs from the two passes are concatenated together using $h_t = [\vec{h}_t; \overleftarrow{h}_t]$, where \vec{h}_t and \overleftarrow{h}_t denote the output from forward and backward pass respectively and h_t is the final output from the bi-directional network.

5.4.3 Conditional Random Field

Conditional Random Fields (CRF) are a class of models that considers contextual information when making predictions. A CRF classifier was used as the last layer of the network to jointly model the label sequence such that the prediction of each label depends on its contextual labels instead of decoding each label independently [209]. For an input sequence $X = (x_1, x_2, \dots, x_n)$, x_t denotes the vector for the t_{th} word. The output h_t from the bi-directional LSTM layers were first mapped to the labels' space using a feed-forward neural network layer. P_θ denote the mapped predictions from the bi-directional LSTM network where θ is used to represent all network parameters for simplicity purposes. $[P_\theta]_{t,k}$ corresponds to the prediction of assigning the k_{th} label to the t_{th} word in a sequence. The CRF layer maintains a transition score matrix as parameters, which records the likeliness of transitioning from one label to another for two consecutive input words. Combining the CRF transition scores with the predictions from the bi-directional LSTM network, the final score for a sequence of labels $y = (y_1, y_2, \dots, y_n)$ is computed using Equation 5.1.

$$s(X, y, \tilde{\theta}) = \sum_{t=0}^n A_{y_t, y_{t+1}} + \sum_{t=1}^n [P_\theta]_{t, y_t} \quad (5.1)$$

where $\tilde{\theta} = \theta \cup \{[A]_{ij} \forall i, j\}$ represents the new model parameters and \mathbf{A} is the matrix of transition scores where $[A]_{ij}$ represents the transition score from label i to label j . The probability of a label sequence over all possible label sequences is computed using a softmax function [178] as presented in Equation 5.2.

$$p(y|X, \tilde{\theta}) = \frac{\exp(s(X, y, \tilde{\theta}))}{\sum_{y' \in Y_X} \exp(s(X, y', \tilde{\theta}))} \quad (5.2)$$

where Y_X denotes the set of all possible label sequences and y' denotes one possible label sequence in that set. The training process maximizes the log-likelihood of the correct label sequence as presented in Equation 5.3.

$$L(\tilde{\theta}) = \log(p(y|X, \tilde{\theta})) \quad (5.3)$$

5.4.4 Bi-directional-LSTM-CRF

The sequence labeling model is ultimately constructed by feeding the output vectors from bi-directional LSTM network into the CRF layer. The output from word embedding for each sentence from the inspection reports were first fed into the bi-directional LSTM network. The output vectors were then passed to the CRF layer to be jointly decoded into a sequence of labels. In this way, the sequence labeling model assigns each sentence from the inspection reports with a sequence of labels. In each sentence, the consecutive words with the same label were then extracted as a connected chunk of bridge condition.

5.5 Data Collection and Preparation

5.5.1 Inspection Report Corpus

A collection of bridge inspection reports were obtained from the Virginia Department of Transportation (VDOT) database. These reports documented bridge conditions as assessed by the VDOT inspection personnel during bridge field inspections in the form of natural language descriptions. Figure 5.4 illustrates the number of words for the three components (deck, superstructure, and super-

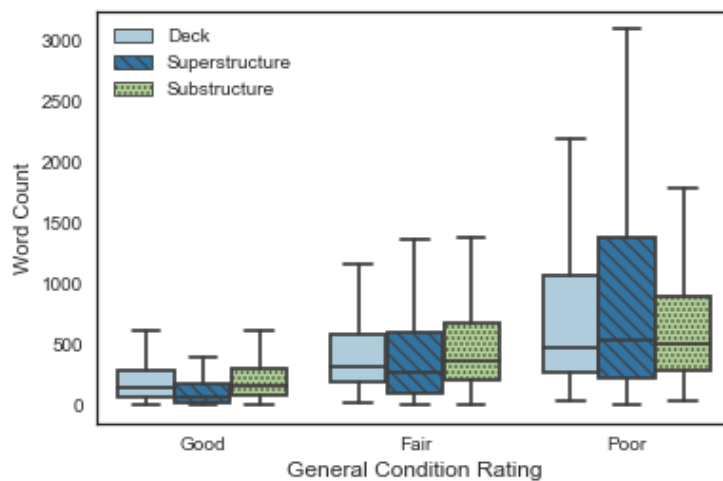


Figure 5.4: Boxplot of number of words documenting the deck, superstructure and substructure components for over 21,000 bridge inspection reports collected from Virginia Department of Transportation (VDOT).

structure) of over 21,000 bridge inspection reports with different general condition ratings (good, fair, and poor denote condition rating values greater than 6, equal to 5 or 6, and less than 5, respectively). As shown in Figure 5.4, inspection reports for bridges in worse condition tend to include more words (sometimes on the order of a few thousand words to describe a component), which indicates the level of detail for the information buried in the inspection reports. Furthermore, it can be seen that the average number of words used to describe each of the three components in an inspection report is close in each condition category, showing the relatively consistent documentation of the conditions of all components.

The inspection reports used in this study were originally in raw document format, organized by the section headers of ‘Deck’, ‘Superstructure’, ‘Substructure’, etc. To obtain a clean inspection report corpus, the textual descriptions were first extracted from the inspection report documents, and irrelevant details were excluded such as table frames, document headers and footers, images and their captions. The textual descriptions were then tokenized into individual tokens, which are contiguous characters between two spaces, usually words. The tokenization was performed using the word tokenizer from the Natural language Toolkit (NLTK) [177], which also splits standard contractions and separates most punctuation characters as separate tokens. The tokens were then joined by a white-space and fed through the sentence tokenizer from NLTK to be organized into sentences. The sentence tokenizer from NLTK was pre-trained using a large corpus to identify the boundaries of the sentences [177]. Tokens from all sentences were reduced to lowercase to be further used for information extraction tasks.

5.5.2 Ground Truth Labeling Categories

Manually annotated ground truth of labeling categories were needed for both training and evaluating the sequence labeling model. For the manual annotation, thirty inspection reports (approximately 12,000 tokens (individual words)) were randomly sampled from the pool of inspection reports for this study from nine different VDOT district offices creating a considerable variety in the training, validation, and testing data. It should be noted that determining the required amount of annotated textual data for a desired level of performance requires further investigation and sensitivity analysis that requires appropriate resources for manual annotation and should be addressed in future works. Labels of the categories were manually assigned, which helped describe the state of bridge damage conditions. The labeling categories include condition Name (N), Severity (S), Location (L) and

Other (O). The 'Other' category contains words connecting the name, location or severity chunks that are not relevant to bridge conditions. To further categorize the severity descriptions, qualitative severity descriptions and numerical severity descriptions were labeled as two separate categories (S(Q) and S(N), respectively). The preprocessed sentences and their tokens were saved using plain text files, where each line contains a token and its label, separated by a white space, and an empty line is used between each two sentences.

5.5.3 Word Embedding

This study used the pre-trained word embeddings developed by [210] as initial word representations, which were then fine-tuned to the inspection domain-specific data during training. These embeddings were developed by training a variant of the skip-gram model on the English Wikipedia August 2015 collection of 2 billion words. The skip-gram model used neighboring and grammatically closer words to predict a target word, and has been proven to encode both semantic and grammatical information in the embeddings [211] that results in a better performance in sequential tasks [210], and therefore was selected as the initial word representations in this study. These embeddings were 300-dimensional ($N = 300$) with a vocabulary size $V = 222,310$ words.

To examine how efficiently these embeddings represent the semantic meanings and contexts of words, word distances between a few selected words and every other word in the vocabulary were computed in the space of embedding vectors using the cosine similarity as presented in Equation 5.4.

$$\text{cosine_similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (5.4)$$

Figure 5.5 presents the T-distributed Stochastic Neighbor Embedding (t-SNE) [212] visualization of the selected words and the words that were most similar to them based on cosine similarity of the word embeddings. T-SNE is a method for dimensionality reduction to reduce multi-dimensional data to a few dimensions suitable for human observation [212]. This method maps the data into a low-dimensional space by minimizing the differences between the probabilities of similarity of points in the two spaces [212]. As shown in Figure 5.5, the words computed as the most similar to the selected words do have similar semantic meanings and context, for example, the most similar words to 'rebar' were steel, I-beams, rivets, girders, wire and etc.; the most similar words to 'corrosion' were abrasion, breakage, embrittlement, passivation, spalling and etc. It was also shown in Figure 5.5 that the clusters of similar selected words were closer to each other such as clusters repre-

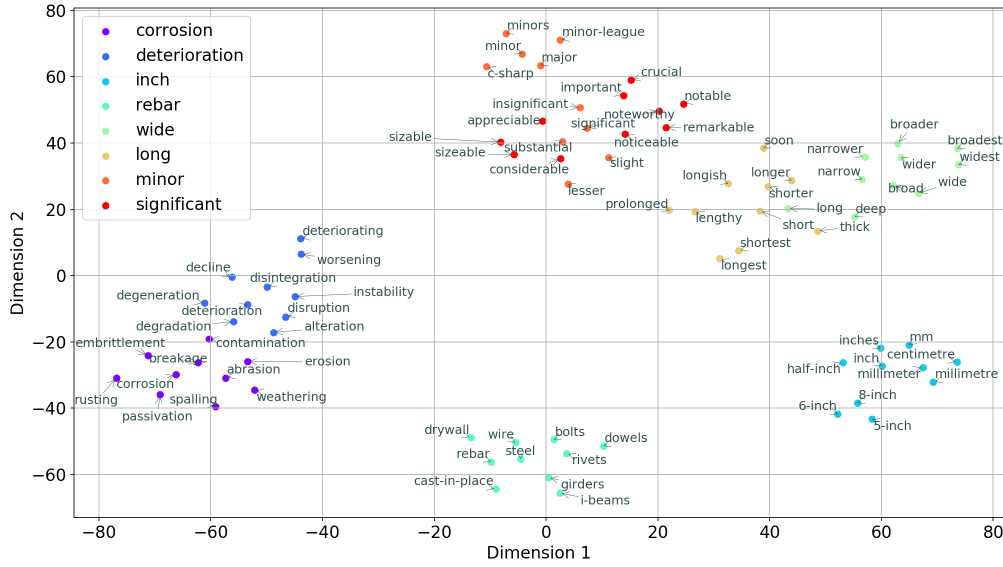


Figure 5.5: T-SNE plot of the selected words and the words that were most similar based the word embeddings.

senting defect types (e.g. ‘corrosion’ and ‘deterioration’), those representing severity (e.g. ‘minor’ and ‘significant’), and those indicating extent (e.g. ‘long’ and ‘deep’). This outcome reinforces the premise that the dependency-based embeddings used are able to encode the semantic and contextual information of the words often used to describe structural deficiencies in the inspection reports.

While the pre-trained embeddings were trained using large Wikipedia corpus and contain a large vocabulary, some words from the inspection report corpus might still not be included in the vocabulary of the pre-trained embeddings. In the case of words that were out-of-vocabulary (OOV) of the pre-trained embeddings, the following procedure was implemented: the numeric characters in an OOV token were first replaced by a special NUMBER token for a re-check. If the token was still not in the vocabulary, it was then added to the vocabulary with randomly-initialized word embeddings if the following two criterion were satisfied: the token appeared at least 5 times in the training data and it contains only letter or number characters. If either one of these criterion was not satisfied, the token was then replaced by a special UNKNOWN token. The two criterion were decided based on a manual examination of all the OOV words (395 tokens in total). The majority of such words appeared less than 5 times in the training corpus (typically only once), and were usually associated with a mix of numbers and special characters for measurement descriptions such as 1

1/4", and 1'0", missing space between words that led to two words being tokenized as one token, and data entry errors such as misspelled words. The first situation is rooted in the fact that bridge inspection reports are free-form textual data that contain rich technical details. For example, it is typical to use ' for feet and " for inch in bridge inspection reports. In consideration of keeping these technical details, no harsh punctuation was performed when preparing the text data before model training. While it is a difficult task to define regular expression matching rules to address the punctuation and special character usage, it is believed that this sequence labeling model will help alleviate the complexity of normalization by dividing the tokens into condition categories. Nine OOV words matched the two criteria explained above, which were not a part of the embedding vocabulary but were somewhat frequent in the inspection report corpus. These words were added into the embedding vocabulary with randomly-initialized embedding weights. These words include : breastwall, spalls, wingwall, backwall, delaminated, wingwalls, spalled, midspan, and fullheight. These words are very unique to the field of bridge engineering and thus did not appear in the vocabulary of general pre-trained embeddings. The fact that only nine frequent words were OOV demonstrates the comprehensiveness of the embedding vocabulary. The pre-trained embeddings were allowed to be further fine-tuned during the gradient updating of the neural network, which adjusts the embeddings to the application of labeling the inspection reports. The effectiveness of this approach has been explored in previous sequential labeling tasks [69, 77].

5.6 Results and Discussions

The inspection report corpus was randomly split into training, validation and testing sets with a ratio of 8:1:1 in terms of number of sentences. Ten times of random split was performed, which generated ten combinations of training, validation, and testing sets, to support comprehensive evaluation of the proposed model and the baseline models using the mean and standard deviation of performance metrics. Table 5.2 presents the statistics of one split of the inspection report corpus into training, validation, and testing sets. For each split, the training set was used for model training iterations, while the validation set was used to evaluate the performance of the model after each training iteration. The evaluation of the model on the validation set guides the decisions on model development, such as hyper-parameter settings. The testing set was held unseen during model development to allow for an objective evaluation of the performance of the model once finalized.

The neural network model was implemented using Keras with Tensorflow [213] as the backend.

Table 5.2: Statistics of one split of the inspection report corpus into training, validation, and testing sets.

Corpus	# Sentences	# Tokens	% Name (N)	% Lo- cation (L)	% Quali- tative Severity (S(Q))	% Numerical Severity (S(N))	% Other (O)
Training	917	9,356	15.81	31.49	5.39	28.36	18.97
Validation	115	1,281	16.63	32.16	5.39	26.93	18.89
Testing	115	1,165	15.11	31.85	6.52	27.64	18.88

The network was trained using the mini-batch stochastic gradient descent (SGD) approach, which minimizes the negative log-probability (Equation 5.3) by iteratively updating the model parameters by one step along the parameters’ gradients based on small subsets of training data (mini-batches). Nesterov-accelerated adaptive moment estimation (Nadam) was used to regulate the training process as recommended by [214] for faster and improved convergence. The batch size was set as 8 and a learning rate of 0.001 was experimentally selected by tuning this parameter. Further trial and error with these parameters did not provide significant performance improvements. To alleviate the exploding gradient issue (gradients with extremely large values) [215] during the training of LSTM networks, gradient normalization [216] was adopted with a threshold $\tau=1.0$ that re-scales the gradient g by a factor $\tau/\|g\|$ when the norm $\|g\|$ exceeds the threshold. Early stopping strategy [217] with a patience parameter of 5 was used to avoid overfitting while training the neural network, where training iterations were terminated once no improvement was observed for 5 consecutive epochs on the validation set. Two layers of stacked LSTM were used, each with a size of 100 recurrent units, as recommended in [214]. Further tuning the number and size of the LSTM layer did not result in significant changes in model performance. The model developed from this study is available online ¹. Table 5.3 presents the summary of hyper-parameter selection.

Model predictions were compared with the ground truth labeling categories for the evaluation of model performance. Precision, recall and F1 score were selected as the evaluation metrics, as

¹<https://github.com/tl6kk/bridge-report-tagging>

Table 5.3: Hyper-parameter settings for the modeling pipeline used in this study.

Parameter Name	Parameter Value
LSTM layers	2
LSTM size	100
Batch size	8
Learning rate	0.001
Gradient normalization	1.0
Early stopping	5

presented in Equations 5.5, 5.6, 5.7.

$$Precision_i = \frac{N_{ii}}{\sum_j N_{ji}} \quad (5.5)$$

$$Recall_i = \frac{N_{ii}}{\sum_j N_{ij}} \quad (5.6)$$

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (5.7)$$

where N_{ij} denotes the number of tokens that belong to class i but were predicted as class j . Precision evaluates the percentage of correctly assigned labels in all labels assigned, while recall evaluates the percentage of each category that were correctly identified. Precision and Recall help demonstrate the full scope of the performance as one focuses on how pure the predictions of each class are (precision), and the other shows how successful the model is in finding all instances of a class (recall). F1 score summarizes the performance in the form of an overall metric and is especially critical when the classes are imbalanced.

The above performance computations measure the performance of the model at the level of the tokens by calculating the metrics based on the number of correctly and incorrectly predicted tokens. The token-level metrics evaluate the model’s ability to capture each word’s context while assigning a condition category to that word. It should also be noted that Precision, Recall, and F1 score can also be evaluated at the chunk level, where the N_{ij} in the chunk-level metrics denotes the number of chunks that belong to class i but were predicted as class j . Denoting a chunk as a group of tokens with the same condition category, a chunk is considered correctly labeled only if every token in the chunk as well as the boundary of the chunk are correctly predicted. Chunk-level evaluation links directly to engineering practice as it reveals the quality of the extracted condition chunks. A similar

evaluation can be conducted at the sentence level, where a sentence is considered correct only if every token in the sentence is labeled the same as in the ground truth. In this regard, sentence-level accuracy can be calculated as the number of correctly predicted sentences divided by the total number of sentences. It should be noted that Precision, Recall, and F1 score cannot be evaluated at the sentence level since the mis-predicted tokens in one sentence can belong to different types, e.g., some Location tokens may be predicted as Condition Name or Severity in the same sentence. Additionally, the sentence-level metric is particularly strict as it counts a sentence as incorrect even if the majority of the token have been correctly predicted. As sentences are not used as a unit of information extraction in the proposed application, this metric is calculated and reported herein solely for sensitivity analysis and comparison purposes.

5.6.1 Model Performance

The performance of the proposed Bi-LSTM-CRF model on the training, validation and testing sets is presented in Table 5.4. The proposed model achieved an F1 score of 94.12% when examined on testing sets, which demonstrated that the model can successfully categorize inspection report text that are unseen during the training process. The training and testing performance did not demonstrate significant gap (5.26%) which shows proper training with limited overfitting.

Table 5.4: Training, validation and testing F1 score for the proposed Bi-LSTM-CRF model.

Bi-LSTM-CRF (Proposed)	Training		Validation		Testing	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
	99.38	00.77	94.13	1.99	94.12	0.52

To further analyze the performance of the proposed Bi-LSTM-CRF model, Table 5.5 presents the confusion matrix for all tokens in the ten randomly-split testing sets, where the presented numbers are the summation of the ten testing sets. Each column of the confusion matrix represents the number of tokens in a predicted class while each row represents the number of tokens in the actual class. As shown in Table 5.5, the model is very accurate in identifying qualitative severity S(Q) descriptions, which is likely due to the fact that the qualitative severity category contains mostly words unique to this class such as 'significantly' and 'light'. The model made the most mistakes in identifying the 'other' category, which is associated to the fact that the 'Other' class is a highly non-homogeneous set that contains all the words that do not belong to the other four defined categories

Table 5.5: Confusion matrix for all tokens in the ten randomly-split testing sets. (N: Name, L: Location, S(Q): Qualitative severity, S(N): Numerical Severity, O: Other)

Testing Set	Predicted: N	Predicted: L	Predicted: S(Q)	Predicted: S(N)	Predicted: O
True: N	1528	60	2	52	38
True: L	86	3023	4	83	71
True: S(Q)	3	1	575	3	7
True: S(N)	56	72	1	2718	22
True: O	84	89	4	58	1943

and therefore might have less unique features for the model to capture.

Figure 5.6 presents a closer examination of sample sentences with mispredicted categories. The sentences are shown with color-coded categories, where yellow, purple, cyan, and white represent condition name, location, severity (numerical) and other categories, respectively. For each example, the sentence with ground truth condition categories is presented first, with the predicted categories presented in the row below. Overall, it can be seen that although the textual data from bridge inspection reports is in the natural free form resulting from the inspectors taking notes during field inspections, the model was able to capture the information from the context and correctly assign a category to most of the words except for a few error cases. Both Examples 1 and 2 resulted from the confusion between location and numerical severity. In Example 1, *approximately 50'-0"* describes an exact location using a numerical measurement in the form of a relative location that was confused with numerical severity. Example 2 demonstrates a relatively harder example for category labeling since several severity descriptions (*2-1/8"*, *1-3/8"*, *1/8"*) and location descriptions (*left wing*, *at the top*, *at the bottom*) were coupled together. Examples 3 and 4 present the error cases regarding the confusion between condition name and numerical severity. In Example 3, the word *up* is mis-labeled as a numerical severity, which may be attributed to the frequent use of the preposition "up" together with measurements as a numerical severity indicator (e.g. *up to 6'-0"h* in Example 1). Example 4 illustrates a miss by the model in recognizing numerical severity. Example 5 and 6 demonstrate error cases regarding the confusion between condition name and location. The misprediction of *in contraction* as location is likely due to the preposition *in* usually being a strong indicator of location descriptions. The sentence in Example 6 includes multiple types of mispredictions. This error can

Example 1	True	embankment erosion up to 6'-0" h with exposed roots, approximately 50'-0" upstream from barrel 1
	Predicted	embankment erosion up to 6'-0" h with exposed roots, approximately 50'-0" upstream from barrel 1
Example 2	True	left wing pushed forward 2-1/8" at the top to 1-3/8" at the bottom and separated 1/8"
	Predicted	left wing pushed forward 2-1/8" at the top to 1-3/8" at the bottom and separated 1/8"
Example 3	True	Asphalt cracked and breaking up 3" wide x 1" deep over abutments
	Predicted	Asphalt cracked and breaking up 3" wide x 1" deep over abutments
Example 4	True	1'-0" silt built up in upstream channel
	Predicted	1'-0" silt built up in upstream channel
Example 5	True	Abutment A - Beam 1 downstream anchor bolt is jammed in contraction
	Predicted	Abutment A - Beam 1 downstream anchor bolt is jammed in contraction
Example 6	True	36' long hairline crack along bottom of upper fillet extending to 1st vertical hairline crack
	Predicted	36' long hairline crack along bottom of upper fillet extending to 1st vertical hairline crack

Condition Name
Location
Severity (Numerical)
Other
mispredicted

Figure 5.6: Examples sentences with mispredicted categories.

be attributed to the fact that the sentence has a rare and complicated usage of other damages as the reference of location. The mention of *1st vertical hairline crack* is used to describe the location of the crack described in the beginning of this sentence, but the model failed to recognize such nuanced usage and mispredicted the labels of the last four words. This examination of the error cases highlights that the textual data from bridge inspection reports are highly heterogeneous in the form of natural language, and the fact that the model achieved an average 94.12% F1 score during testing with limited errors on complex sentences proves the feasibility of the proposed approach.

5.6.2 Effect of Context Awareness

This study focuses on automated information extraction from bridge inspection reports using deep-learning-based (DL-based) approach. The proposed system enforces the context-awareness required for effective bridge management information extraction by incorporating context-aware components including dependency-based word embeddings, bi-directional LSTM cells, and Conditional Random Fields classifier on top of a regular RNN network. Table 5.6 presents the effect of the incorporated context-aware components in the proposed system compared with baseline models. Each model was trained and evaluated using the same ten combinations of training, validation, and testing set. Performance of the models is presented using means and standard deviations computed using the metrics defined in Equation 5.5, 5.6, 5.7.

As presented in Table 5.6, mainly two families of models were compared: baseline models with-

Table 5.6: Performance of different model architecture variants compared with baseline models (token leve).

Model Family	Model	Precision		Recall		F1 score	
		Mean	Stdev	Mean	Stdev	Mean	Stdev
Non-context Baselines	SVM (feat)	86.57	1.18	51.14	3.25	55.55	3.91
	SVM (w2v)	82.54	1.75	80.57	2.33	81.32	2.16
	RNN (w2v)	89.46	1.35	88.68	1.47	89.07	1.35
Context-aware Models	RNN (dep)	91.28	1.55	90.45	1.50	90.86	1.50
	LSTM (dep)	92.40	1.11	92.05	1.02	92.22	1.02
	Bi-LSTM (dep)	94.04	0.74	93.66	0.78	93.85	0.70
	Bi-LSTM-CRF (dep)	94.31	0.55	93.94	0.57	94.12	0.52

out the context-aware components (Non-context baselines) and the proposed model with context-aware considerations (Context-aware models). The non-context model family includes two ML baselines and a DL baseline model. Both ML baseline systems, SVM (feat) and SVM (w2v), used Support Vector Machine (SVM) [218] models. The two SVM baseline systems were set up as follows: both systems tag a current word using features from its previous word, following word, and the current word itself (similar to [39]). The SVM (feat) was trained using twelve-dimensional features. Four features were generated for each word: is_digit, is_letter, the Part-of-Speech tag (grammatical categories such as noun, verb, etc.). To predict the condition category of a current word, the four features from its previous word, itself and the word after it were concatenated together. The SVM (w2v) used word embeddings previously trained using the skip-gram model [70], denoted as w2v. The embeddings for the previous word, the current word and the word after were concatenated together for the prediction of the current condition category. As seen in Table 5.6, the proposed model significantly outperformed the two SVM baseline systems across all three metrics. The poor performance of the SVM-feat can be attributed to the fact that neither the model nor the features exploited the dependencies and contexts of word sequences. SVM-embed achieved improvements compared to SVM-feat because the word embedding encoded a certain level of contextual information. The DL baseline, RNN (w2v) significantly improved the performance in terms of precision, recall, and F1 score comparing with the ML baselines as presented in Table 5.6, which illustrated the capability of DL-based methods in integrating dependencies among sequential inputs. The RNN (w2v) model

was initialized using word embeddings previously trained using the skip-gram model [70], which are the same embeddings as used in SVM (w2v).

The model performance of the context-aware family further revealed the effectiveness of each context-aware component in the proposed model. The context-aware model family includes the proposed model as well as three of its variants: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) [84], and bi-directional LSTM (Bi-LSTM). The effectiveness of each component of the proposed model are illustrated by the incremental improvements between each two consecutive rows in Table 5.6. All the context-aware models were trained using dependency-based word embeddings [210] fine-tuned to the existing dataset (denoted by (dep) in Table 5.6), and the hyperparameters were set according to Table 5.3. Compared with RNN (w2v), RNN (dep) demonstrated improvements in all three metrics, which shows the benefit of dependency-based embedding in encoding both grammatical and semantic information into word embeddings for sequence labeling tasks. The comparison between RNN (dep) and LSTM (dep) indicates that using LSTM cells slightly improves the performance of RNN across all metrics. Comparing Bi-LSTM (dep) with LSTM (dep) shows that incorporating bi-directional information further improved the performance of the sequence labeling task. Ultimately, the proposed method (Bi-LSTM-CRF) achieved the highest scores across all metrics with all the proposed context-aware components combined.

Figure 5.7 presents examples segments from bridge inspection reports where the same word belongs to different condition categories based on different contexts, and the categorization results from baseline SVM (w2v), RNN (w2v), as well as the proposed system. As shown in Figure 5.7:

- The first set of examples focus on different contexts regarding the word ‘hole’, which is commonly used when describing a defect or a bridge element (*e.g.* weep hole). The word ‘hole’ should be categorized as Condition Name in example 1-4, and as Location in example 5-6. The SVM (w2v) failed to distinguish the word ‘hole’ from its direct neighboring words in example 1, 2, and 5, and mispredicted it as Other in example 4. The RNN (w2v) correctly categorized the word ‘hole’ in all six examples, but failed in categorizing some of its contexts. All three systems failed in example 6, which describes a relatively rare bridge condition of tree root overgrown. The SVM (w2v) and RNN (w2v) obtained broken pieces instead of continuous chunks, while the proposed system mis-labeled the entire chunk as Numerical Severity.
- The second set of examples were related to different contexts of the words ‘girder’ and ‘sur-

Word	#	Ground Truth	SVM (w2v)	RNN (w2v)	Proposed
hole	1	scour hole at outlet	scour hole at outlet	scour hole at outlet	scour hole at outlet
	2	up to 1" diameter hole	up to 1" diameter hole	up to 1" diameter hole	up to 1" diameter hole
	3	corrosion hole at end of beam	corrosion hole at end of beam	corrosion hole at end of beam	corrosion hole at end of beam
	4	lift holes with exposed embedded pipe end	lift holes with exposed embedded pipe end	lift holes with exposed embedded pipe end	lift holes with exposed embedded pipe end
	5	vertical crack at upstream weep hole and is exposed up to 26 inch	vertical crack at upstream weep hole and is exposed up to 26 inch	vertical crack at upstream weep hole and is exposed up to 26 inch	vertical crack at upstream weep hole and is exposed up to 26 inch
	6	tree root growing out of weep hole	tree root growing out of weep hole	tree root growing out of weep hole	tree root growing out of weep hole
girder surface	7	girder 6 two cover spalls	girder 6 two cover spalls	girder 6 two cover spalls	girder 6 two cover spalls
	8	freckled surface rust on less than 5% of girder surface area	freckled surface rust on less than 5% of girder surface area	freckled surface rust on less than 5% of girder surface area	freckled surface rust on less than 5% of girder surface area
	9	medium scale to 60% surface area	medium scale to 60% surface area	medium scale to 60% surface area	medium scale to 60% surface area
	10	surface rust on bearing devices	surface rust on bearing devices	surface rust on bearing devices	surface rust on bearing devices
3'	11	spall 3' wide full height	spall 3' wide full height	spall 3' wide full height	spall 3' wide full height
	12	expansion joint at abutment a totaling 3' was 2'	expansion joint at abutment a totaling 3' was 2'	expansion joint at abutment a totaling 3' was 2'	expansion joint at abutment a totaling 3' was 2'
	13	tree roots exposed 3' behind columns	tree roots exposed 3' behind columns	tree roots exposed 3' behind columns	tree roots exposed 3' behind columns
	14	breastwall starting 3' from upstream end x 8' long	breastwall starting 3' from upstream end x 8' long	breastwall starting 3' from upstream end x 8' long	breastwall starting 3' from upstream end x 8' long

Condition Name
Location
Severity (Numerical)
Severity (Qualitative)
Other
mispredicted

Figure 5.7: Comparison of model prediction examples between non-context and context-aware models.

face’. In example 7, all three systems correctly identified ‘girder’ as Location; similarly, in example 9 and 10, all three systems correctly identified ‘surface’ as Numerical Severity and Condition Name, respectively. However, in example 8, ‘girder surface area’ together was used as the unit of measurement, but the SVM (w2v) and RNN (w2v) failed to categorize some words from this context.

- The third set of examples present the word 3’ which is commonly used in descriptions of Location and Numerical Severity. In example 12, the SVM (w2v) did not correctly identify ‘totaling 3’ as Numerical Severity; while in example 14, all three systems have mispredicted cases. The SVM (w2v) mispredicted ‘3’ from’ as Numerical Severity, and failed to recognize the connection among ‘from upstream end’. The RNN (w2v) also failed to recognize 3’ but mispredicted it as Other category. It should be noted that the proposed model also mispredicted the word ‘end’ as Numerical Severity, which might be attributed to the fact that the description of Numerical Severity is commonly in the form of ‘3’ wide x 3’ long’ (using 3 to represent any measurement number).

Comparing the example predictions from all three systems, both baseline methods failed to capture

the contexts of some example cases, and mispredicted the category of the word when affected by the word’s direct neighboring words, or the word’s commonly-used category in other contexts. Successful context-aware information extraction from bridge inspection reports requires a model to produce continuous chunks of the same category, and meanwhile also correctly break a chunk when there should be a transition of category. The proposed system can successfully capture most of the contextual information and produce predictions that segment sentences into continuous and meaningful chunks. Table 5.7 presents chunk-level and sentence-level performance of the proposed model compared with the two baselines: SVM(w2v) and RNN(w2v). The metrics in this table were calculated as discussed in Section 6. As can be seen in this table similar to Table 7, the proposed Bi-LSTM-CRF model consistently outperformed the two baselines at both the chunk and sentence levels, which further confirms the advantage of the proposed approach. A second conclusion from the evaluations at the three levels is that the overall performance decreases with an increase in the number of tokens taken as the unit of evaluation (i.e. token-level and sentence-level metrics have the highest and lowest values, respectively). This is to be expected as achieving high chunk and sentence-level accuracies requires larger numbers of tokens to be predicted accurately. Finally, it can be observed from the comparison of Tables 7 and 8 that higher levels of evaluation even more strongly highlight the advantage of the proposed method in comparison with the baselines. This is evidenced by the higher gap between the sentence-level (and chunk-level) results of the proposed method when compared with the token-level results.

Table 5.7: Chunk-level and sentence-level performance of the proposed model compared with the two baselines: SVM(w2v) and RNN(w2v).

Model	Chunk Level						Sentence Level	
	Precision		Recall		F1 score		Accuracy	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
SVM(w2v)	55.57	2.37	57.72	2.78	56.24	2.53	49.22	2.69
RNN(w2v)	63.75	3.07	70.23	2.09	66.56	2.55	51.65	5.52
Bi-LSTM-CRF(dep)	81.83	2.22	83.73	2.00	82.72	1.90	71.65	3.81

5.6.3 Analysis of Labeled Inspection Reports

As demonstrated in this paper, the proposed pipeline assigns sequences of labels to the sentences from bridge inspection reports and extracts segments with consecutive labels. The extracted segments then form a structured bridge condition inventory that are organized by the condition categories. Constructing such a fine-grained and comprehensive condition inventory benefits bridge maintenance and management by providing rich, historical information to support big data analytics. This condition inventory can be used to assist in creating targeted insights for infrastructure owners, managers and maintenance planners such as "What are the frequent damage types within a region in Virginia?", or "How many bridges have cracks developed since last inspection and are there severe deterioration or significant changes that warrant warnings?".

In addition to answering such questions, deterioration forecasting models can also be created by tracking and modeling the defects as well as their quantitative severity extracted and structured from the raw inspection reports of one or more structures over the history of the inspections recorded in a DOT database. Figure 5.8 presents an example of using the proposed model to evaluate deterioration in condition descriptions of a concrete slab bridge built in 1932 in Virginia. This table compares condition descriptions for the deck and curb between the inspection reports created by inspectors in 2014 and 2017. Although the same general condition rating "4" was assigned to this

Year	Damage	Location	Severity	Deterioration?
2014	spalling and delamination	bottom of deck downstream side	a 47" long x 29" wide x 2 3/4" deep area	NO
2017				
2014	exposed longitudinal bars	bottom of deck downstream side	4 (rebars)	YES
2017			5 (rebars)	
2014	exposed transverse bar	bottom of deck downstream side	75% (section loss)	YES
2017			75% to 100% (section loss)	
2014	scaling	upstream curb on the inside and outside edges	1 1/4" deep	YES
2017			1 1/4" deep full length	
2014	random cracking	downstream curb on outer face	hairline	NO
2017				
2014	scaling	on top inside edge	1/2" deep	YES
2017		on top inside and outside edge	1" deep 3" long	

Figure 5.8: Condition changes between two inspection dates for a 1932 slab bridge in Virginia.



(a) Exposed rebar and spall on deck bottom (2014).



(2017).

Figure 5.9: Visual illustration of deterioration progress revealed from historical inspection reports.

bridge deck in both inspections, details of deterioration progress can be revealed by the information extracted from inspection reports. As shown in the comparison, the exposed rebar has changed in both quantity (from 4 rebars to 5) and extent of section loss (from 75% to 75-100%). The severity of deck curb scaling has also developed to full length from 2014 to 2017. Furthermore, new scaling has been developed on the outside edge, and the size of scaling has also increased (from 1/2" deep to 1" deep 3" long). Figure 5.9 depicts images from the 2014 and 2017 inspections of the studied slab bridge showing exposed rebars and spalling on the deck bottom. As can be seen, the increase in exposed rebar (the addition of the fifth exposed rebar in 2017) is hardly visible in the 2017 image, and therefore these images do not fully document the progression of damage on their own. However, the extracted information from the textual inspection data is therefore helping to provide detailed condition information that is otherwise hard to obtain and use for the purpose of deterioration monitoring and maintenance decision making.

Chapter 6

Conclusion and Future Work

This study envisions a path from current bridge management practices advancing towards a fully automated, smart bridge infrastructure management system, as presented in Figure 6.1. The current bridge management practice relies on visual inspections to routinely collect bridge condition data. Inspection personnel assign condition ratings and determine repair needs following each inspection. The identified limitations of the current practice are that, ensuring consistency in the experience-based condition rating process is challenging and requires comprehensive training and quality control process; with no detailed local condition data available, maintenance planning is limited to local optimal as determined by local agencies instead of system-level optimization. Meanwhile, the bridge inspection reports generated through years of practice are rich in condition details and condition assessment expertise, which motivates this dissertation to construct automated condition rating and information extraction models using the unstructured data from bridge inspection reports. Envisioning an automated bridge management system that has local conditions and defect measurements provided by robotic inspection and image-based defect detection, the automated condition rating task aims to map local conditions to a global assessment. Such model learns from the collective knowledge base of inspectors' rating expertise in order to improve the consistency in condition rating. The information extraction task aims to extract categorized condition information to construct a structured condition inventory, which can then support system-level maintenance decisions.

To the above regards, Chapter 3 focused on two primary gaps in knowledge identified in a comprehensive review of the literature, namely 1) the lack of quality control works via raw condition descriptions rather than human-assigned ratings, 2) the lack of semantics-based language modeling

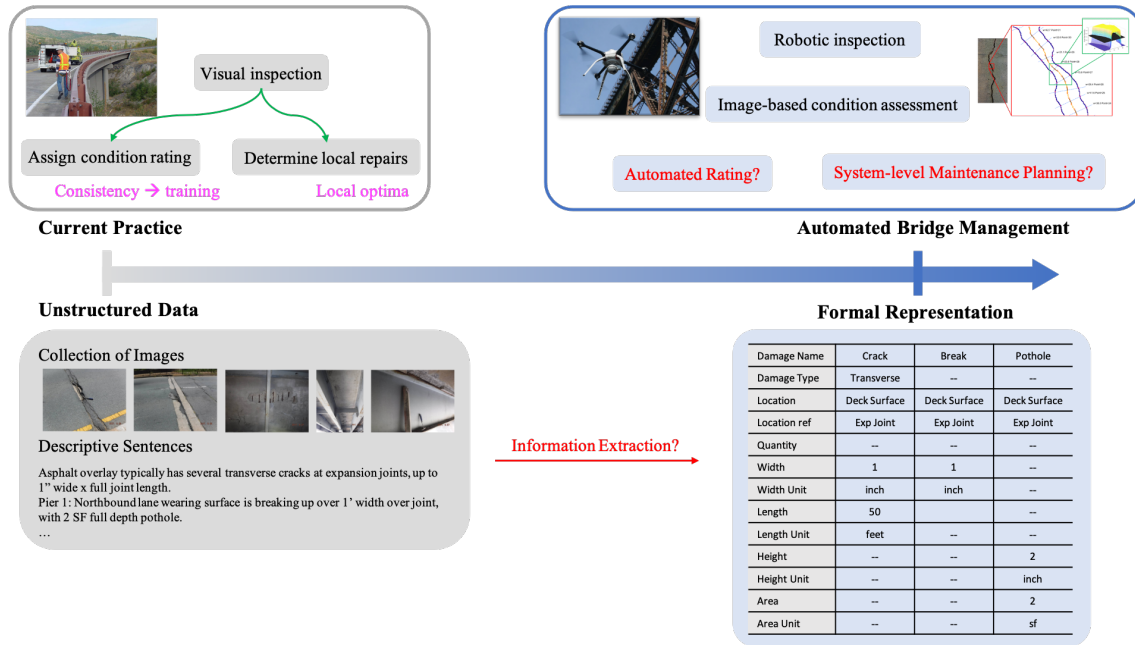


Figure 6.1: Dissertation outline of identified **limitations**, **motivations**, and research **questions**.

using deep learning, especially on ordinal score assignment. These gaps were addressed by collecting textual data from a large amount of bridge inspection reports and constructing a data-driven framework for mapping the narrative descriptions to condition ratings. A hierarchical attention network was developed to fully exploit the semantics and contexts of highly heterogeneous textual data from bridge inspection reports. The proposed framework was developed using bridge inspection reports collected from the Virginia Department of Transportation database.

- The proposed Hierarchical Attention Network outperformed a variety of existing baselines from the literature in the civil infrastructure domain. The testing predictions demonstrated a clear trend concentrating diagonally on the confusion matrix with only 1.25% of predictions missing by more than one level.
- The proposed model is highly interpretable, and the words and sentences emphasized by the model demonstrated trends and insights regarding how local conditions contribute to the overall condition rating. The resulting attention map can also be helpful during the rating process by highlighting potentially important details that may be overlooked by inspectors.
- The proposed quality control process achieved a testing accuracy of 88.67% in deciding to accept or reject synthetic condition ratings. Comparing the likelihood ratio with a data-driven

threshold significantly improved both accuracy and F-1 score compared to direct quality control using model-predicted ratings.

- The data-driven likelihood threshold is affected by the types of errors and the percentage of wrong ratings in the inspector-provided ratings. The threshold can be tuned depending on the less desirable type of error (false positive vs. false negative) to be avoided given engineering judgment in the specific application scenario.
- The presented framework demonstrated feasibility in automated condition rating recommendation, as well as in quality control of bridge condition assessment, and hence can proactively increase the statewide and nationwide consistency of condition rating practices.

Chapter 4 identifies the multi-modal data from bridge inspection reports as a source of domain expertise and proposed a deep learning-based fusion model for automatic condition rating. Each bridge inspection report contains visual data (a collection of images) and textual data (a sequence of sentences) that documents the local condition findings during inspections. The challenges of fusing such two modalities lie in that, visual data provides supplemental information without comprehensively covering every aspect of bridge condition, and each image from the collection does not have alignment with a sentence from the sequence. In that regard, this study developed an image-sentence fusion model that extracts visual and textual representation from images and sentences respectively, and fuses the representations using a sequence encoder followed by an attention mechanism. The model was developed using bridge inspection reports collected from the Virginia Department of Transportation (VDOT) and based on the results of experiments and model uncertainty analysis, the following conclusions can be made:

- The image data alone from bridge inspection reports achieved 49.42% ACC0 for condition rating, while the textual modality performed significantly better than the image, with an ACC0 of 64.43%. The textual data contains more comprehensive information in capturing bridge conditions, while the image data provides supplemental information that does not guarantee the coverage of all local conditions.
- The proposed fusion framework demonstrated further improvement over the uni-modal models for enhanced automated condition rating. As the two uni-modal baselines achieved uneven performance, direct fusion with the uni-modal softmax scores did not demonstrate significant

improvement in rating performance. Allowing image and sentence representation to be jointly encoded and attended reached slightly better results compared with aggregating image and sentence representations separately before fusion.

- With random image augmentation transforms and dropout during testing, the maximum uncertainty in model predictions was 0.817; 95% of the testing data resulted in an uncertainty smaller than 0.535. Considering the estimated uncertainty for inspector-assigned ratings was within two rating points for 95% of the bridge population, the proposed model learns from a large rating database and aggregated inspector expertise in order to alleviate the inconsistency issue in experience-driven bridge condition assessment.
- When applying the automated condition rating model, the consideration of allowable uncertainty help increases overall rating performance, where the predicted rating with exceeding uncertainty may be referred to a human inspector for further investigation. Examining the correlation of model predicted softmax scores revealed that the model tended to not mis-predict outside the neighboring condition categories even in the uncertain setting with random image transforms and dropout.

Chapter 5 extracted condition details from bridge inspection reports in a context-aware framework. The information extraction problem was formulated as a sequential labeling task that processes each sentence from the inspection reports and assigns a condition category label to each word. A Bi-LSTM-CRF model was constructed to enforce the connections amongst the sequential data and assign a label to each word based on its meaning and context. The assigned labels segmented each sentence into chunks of structured condition details including condition names, locations, and (numerical or qualitative) severity. The model was developed using bridge inspection reports collected from the database of the Virginia Department of Transportation. The results provided in this chapter demonstrated the success of the proposed method in extracting deterioration information from bridge inspection reports.

- The proposed information extraction employing a Bi-LSTM-CRF architecture on top of the dependency-based word embeddings achieved powerful performance with an F1 score of 94.12% during testing.
- Each component of the proposed architecture (LSTM cells, bidirectionality, and the CRF

classifier) demonstrated effectiveness for improving the performance of the sequence labeling model.

- The context-awareness enforced through both the architecture and the word embeddings helped improve the performance compared with baselines.
- The extracted condition information was shown to be useful in maintenance planning and decision making by providing targeted insights and historical comparison and change detection.

The results and discussions in this dissertation demonstrated the capability of currently available bridge inspection reports in supporting automated tools, and also pointed to potential future directions towards the wider application of the developed tools, better regularization of future data collection, and eventually the development of a formal representation of bridge conditions that can support detail-oriented, data-driven bridge management systems.

While this study collected inspection reports from the Virginia Department of Transportation (VDOT), a potential future research direction is to extend the proposed condition rating models nationwide with more data collected from other state DOTs and bridge management agencies that aggregate to a comprehensive knowledge base of condition assessment experience and expertise. Depending on the level of variance between the nation-wide reports and those used in this study, the proposed model might need to be fine-tuned on the additional data from other states. Visualization of the resulting model may help reveal the underlying condition assessment patterns for each state. Since each state DOT performs training and quality control separately, having a unified data-driven tool for quality control of the in-training inspectors may improve the consistency in bridge condition assessment nationwide.

As the inspection reports become recognized as a source of condition information for supporting data-driven condition assessment, it is anticipated that future development of a unified guideline or specification of inspection note-taking and report documentation will improve the performance of automated condition rating models. Such guidelines may specify the level of details in inspection notes (*e.g.* describing one local defect with one sentence of its name, location, and severity), the overall length of narrative descriptions, as well as the number and scale of the site images documented in the inspection reports.

The proposed context-aware information extraction model processes bridge inspection reports sentence by sentence and generates segments of bridge condition type, location as well as qualitative

and quantitative severity. One possible future research direction in this regard is to create a detailed national condition inventory by applying the proposed information extraction method to inspection report data from a wide range of states. Such a condition inventory can support data-driven deterioration modeling, and forecasting, which can be an important step in advancing bridge management and maintenance analytics. Further dependency extraction and entity normalization processes might be developed to organize the condition inventory for supporting data-driven decision-making. Although the model-extracted segments can be automatically matched where there is only one segment for each category in the sentence, it can become confusing when multiple segments are from the same category. For a complex sentence that stated two damage conditions, it is not readily clear which location and severity correspond to which damage condition mentioned in the same sentence. Although examination of the inspection report corpus used in this study revealed that only ~13% of the sentences were complex sentences, future work might be required to create relation extraction methods that can match each extracted location and severity segment to their corresponding condition to resolve such ambiguity and confusion. Examples of such methods in the literature developed in different domains can be found [219–221]. Further normalization can be performed for each of the condition categories. For example, the National Bridge Elements [16] can be used to guide the normalization of the location category, to precisely describe the locations in terms of a defined bridge element and a direction indicator such as top or bottom. The structure of the inventory needs to be carefully designed to organize the records of defects by bridge elements and direction indicators, so that the descriptions of the same defects from consecutive years can be aligned and compared. Units and measurements from the severity category can also be further normalized to a unified form to precisely describe the size and extent of each defect.

The development of such a comprehensive national bridge condition inventory links closely to the design of a formal representation for bridge conditions. Such representation is aimed to document all local bridge deficiencies in a tabular database format in order to support detail-driven feature extraction for deterioration modeling. The design of such representation is a non-trivial task that comprehensively identifies the formal representation of all defects, locations, and severity. For example, a non-formal representation of a local defect can be ‘1/3 in. w. crack on the bottom of the deck near abutment 2’, and the formal representation should include defect name (crack), defect type (NA), measurement (0.3), measurement type (width), measurement unit (inch), location (deck), location type (bottom), location reference (mid-span abutment), and *etc.*. Each item in the

formal representation should be coded in numerical format, which requires defining a categorical variable by identifying all possible values of each item. The entire evolutionary history of bridge conditions documented in the inspection reports enables the construction of such representation, and the resulting condition inventory has extensive potential in supporting data-driven deterioration modeling and maintenance planning at an unprecedented level of detail. Such formal representation of bridge condition can also guide the design of the automatic inspection process, where the inspector is only responsible for inputting measurement numbers, quantities, or answers to multi-choice questions that will be further analyzed by automatic condition rating and deterioration prediction models.

Bibliography

- [1] FHWA, Recording and coding guide for the structure inventory and appraisal of the nation's bridges, Rep. No. FHWA-PD-96-001 (1995).
- [2] T. Ryan, J. Mann, Z. Chill, B. Ott, Bridge inspector's reference manual (birm), Arlington, Virginia: US Department of Transportation (2012).
- [3] VDOT, 2020. URL: <http://www.virginiadot.org/travel/parkride.asp>.
- [4] X. Rong, word2vec parameter learning explained, arXiv preprint arXiv:1411.2738 (2014).
- [5] R. G. Mishalani, S. M. Madanat, Computation of infrastructure transition probabilities using stochastic duration models, *Journal of Infrastructure systems* 8 (2002) 139–148.
- [6] M. S. Darmawan, M. G. Stewart, Spatial time-dependent reliability analysis of corroding pretensioned prestressed concrete bridge girders, *Structural Safety* 29 (2007) 16–31.
- [7] C.-A. Robelin, S. M. Madanat, History-dependent bridge deck maintenance and replacement optimization with markov decision processes, *Journal of Infrastructure Systems* 13 (2007) 195–201.
- [8] G. Morcous, Performance prediction of bridge deck systems using markov chains, *Journal of performance of Constructed Facilities* 20 (2006) 146–155.
- [9] Q. Jin, Z. Liu, J. Bin, W. Ren, Predictive analytics of in-service bridge structural performance from shm data mining perspective: A case study, *Shock and Vibration* 2019 (2019).
- [10] H. Zhang, D. W. R. Marsh, Multi-state deterioration prediction for infrastructure asset: Learning from uncertain data, knowledge and similar groups, *Information Sciences* (2019).

- [11] M. Moore, B. M. Phares, B. Graybeal, D. Rolander, G. Washer, J. Wiss, et al., Reliability of visual inspection for highway bridges, volume I, Technical Report, Turner-Fairbank Highway Research Center, 2001.
- [12] L. W. Luther, Code of federal regulations, in: National Bridge Inspection Standards (23 CFR 650), U.S. Government Printing Office, revised April 1, 1998, p. 238–240.
- [13] R. B. Buchheit, J. H. Garrett Jr, S. McNeil, P. Chen, Automated procedure to assess civil infrastructure data quality: Method and validation, *Journal of infrastructure systems* 11 (2005) 180–189.
- [14] G. Migliaccio, S. M. Bogus, A. Cordova-Alvidrez, Continuous quality improvement techniques for data collection in asset management systems, *Journal of Construction Engineering and Management* 140 (2014) B4013008.
- [15] Z. U. Din, P. Tang, Automatic logical inconsistency detection in the national bridge inventory, *Procedia Engineering* 145 (2016) 729–737.
- [16] M. Farrar, B. Newton, M. Johnson, P. Jensen, D. Juntunen, G. Christian, T. Everett, L. Hummel, J. Thiel, W. Casey, *The AASHTO Manual for Bridge Element Inspection*, 1st edition, American Association of State Highway and Transportation Officials (AASHTO), Washington DC, 2010.
- [17] S. Lim, S. Chung, S. Chi, J. Song, A framework for developing an estimation model of damages on bridge elements using big data analytics, in: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, volume 33, Vilnius Gediminas Technical University, Department of Construction Economics . . . , 2016, p. 1.
- [18] K. Chang, S. Chi, Bridge clustering for systematic recognition of damage patterns on bridge elements, *Journal of Computing in Civil Engineering* 33 (2019) 04019028.
- [19] Federal Highway Administration, Highway statistics 2019, Accessed: March 7, 2021, <https://www.fhwa.dot.gov/policyinformation/statistics/2019/>, 2019.
- [20] American Society of Civil Engineers, Report card for america’s infrastructure, <https://infrastructurereportcard.org/wp-content/uploads/2020/12/Bridges-2021.pdf>, 2020.

- [21] American Road and Transportation Builders Association, ARTBA bridge report, <https://artbabridgereport.org/reports/2020%20ARTBA%20Bridge%20Report.pdf>, 2020.
- [22] U.S. Department of Transportation, Federal Highway Administration, Status of the nation's highways, bridges, and transit conditions and performance report: Conditions and performance, 23rd edition, <https://www.fhwa.dot.gov/policy/23cpr/pdfs/pdf/23cpr.pdf>, 2020.
- [23] R. A. Hartle, T. W. Ryan, E. Mann, L. J. Danovich, W. B. Sosko, J. W. Bouscher, M. Baker Jr, et al., Bridge Inspector's Reference Manual: Volume 1 and Volume 2, Technical Report, United States. Federal Highway Administration, 2002.
- [24] E. Grimm, D. Lehmann, Washington state bridge inspection manual, Washington: Washington State Department of Transportation Administrative and Engineering Publications (2010).
- [25] R. Marz, Wisconsin structure inspection manual, Madison, WI (1996).
- [26] S. Mohamadi, D. Lattanzi, Life-cycle modeling of structural defects via computational geometry and time-series forecasting, *Sensors* 19 (2019) 4571.
- [27] S. Mohamadi, D. Lattanzi, A computational geometry approach to the life-cycle modeling of remotely-sensed defects, in: *Computing in Civil Engineering 2019: Smart Cities, Sustainability, and Resilience*, American Society of Civil Engineers Reston, VA, 2019, pp. 531–537.
- [28] D. Lattanzi, G. Miller, Review of robotic infrastructure inspection systems, *Journal of Infrastructure Systems* 23 (2017) 04017004.
- [29] N. Charron, E. McLaughlin, S. Phillips, K. Goorts, S. Narasimhan, S. L. Waslander, Automated bridge inspection using mobile ground robotics, *Journal of Structural Engineering* 145 (2019) 04019137.
- [30] M. N. Gillins, D. T. Gillins, C. Parrish, Cost-effective bridge safety inspections using unmanned aircraft systems (uas), in: *Geotechnical and Structural Engineering Congress 2016*, 2016, pp. 1931–1940.
- [31] N. Gucunski, B. Basily, J. Kim, J. Yi, T. Duong, K. Dinh, S.-H. Kee, A. Maher, Rabbit: implementation, performance validation and integration with other robotic platforms for improved

- management of bridge decks, *International Journal of Intelligent Robotics and Applications* 1 (2017) 271–286.
- [32] A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell’Andrea, M. Riley, Unmanned aerial vehicle inspection of the placer river trail bridge through image-based 3d modelling, *Structure and Infrastructure Engineering* 14 (2018) 124–136.
- [33] S. Phillips, S. Narasimhan, Automating data collection for robotic bridge inspections, *Journal of Bridge Engineering* 24 (2019) 04019075.
- [34] AASHTO, Summary of results from state dot drone usage surveys., Accessed: March 8, 2021, <https://transportation.libguides.com/uav/surveys>, 2018.
- [35] J. Zhang, N. M. El-Gohary, Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking, *Journal of Computing in Civil Engineering* 30 (2013) 04015014.
- [36] J. Zhang, N. M. El-Gohary, Extending building information models semiautomatically using semantic natural language processing techniques, *Journal of Computing in Civil Engineering* 30 (2016) C4016004.
- [37] T. Kim, S. Chi, Accident case retrieval and analyses: Using natural language processing in the construction industry, *Journal of Construction Engineering and Management* 145 (2019) 04019004.
- [38] S.-K. Lee, B. Kim, M. Huh, J. Park, S. Kang, S. Cho, D. Lee, D. Lee, Knowledge discovery in inspection reports of marine structures, *Expert Systems with Applications* 41 (2014) 1153–1167.
- [39] K. Liu, N. El-Gohary, Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports, *Automation in Construction* 81 (2017) 313–327.
- [40] K. Liu, N. El-Gohary, Similarity-based dependency parsing for extracting dependency relations from bridge inspection reports, in: *Computing in Civil Engineering 2017*, 2017, pp. 316–323.

- [41] P. Zhou, N. El-Gohary, Domain-specific hierarchical text classification for supporting automated environmental compliance checking, *Journal of Computing in Civil Engineering* 30 (2015) 04015057.
- [42] K. Liu, N. El-Gohary, Unsupervised named entity normalization for supporting information fusion for big bridge data analytics, in: *Workshop of the European Group for Intelligent Computing in Engineering*, Springer, 2018, pp. 130–149.
- [43] K. Liu, N. El-Gohary, Semantic modeling of bridge deterioration knowledge for supporting big bridge data analytics, in: *Construction Research Congress 2016*, 2016, pp. 930–939.
- [44] Y. Li, H. Li, H. Wang, Pixel-wise crack detection using deep local pattern predictor for robot application, *Sensors* 18 (2018) 3042.
- [45] M. Alipour, D. K. Harris, G. R. Miller, Robust pixel-level crack detection using deep fully convolutional neural networks, *Journal of Computing in Civil Engineering* 33 (2019) 04019040.
- [46] Q. Mei, M. Gül, M. R. Azim, Densely connected deep neural network considering connectivity of pixels for automatic crack detection, *Automation in Construction* 110 (2020) 103018.
- [47] P. Hühwohl, R. Lu, I. Brilakis, Multi-classifier for reinforced concrete bridge defects, *Automation in Construction* 105 (2019) 102824.
- [48] M. Alipour, D. K. Harris, Increasing the robustness of material-specific deep learning models for crack detection across different materials, *Engineering Structures* 206 (2020) 110157.
- [49] P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, H. Parvardeh, Automated crack detection on concrete bridges, *IEEE Transactions on automation science and engineering* 13 (2014) 591–599.
- [50] H. Kim, E. Ahn, M. Shin, S.-H. Sim, Crack and noncrack classification from concrete surface images using machine learning, *Structural Health Monitoring* 18 (2019) 725–738.
- [51] T. Omar, M. L. Nehdi, T. Zayed, Infrared thermography model for automated detection of delamination in rc bridge decks, *Construction and Building Materials* 168 (2018) 313–327.

- [52] Y. Gao, K. M. Mosalam, Deep transfer learning for image-based structural damage recognition, *Computer-Aided Civil and Infrastructure Engineering* 33 (2018) 748–768.
- [53] Y. Xu, Y. Bao, J. Chen, W. Zuo, H. Li, Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images, *Structural Health Monitoring* 18 (2019) 653–674.
- [54] A. Zhang, K. C. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, C. Chen, Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network, *Computer-Aided Civil and Infrastructure Engineering* 32 (2017) 805–819.
- [55] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P. Fieguth, A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure, *Advanced Engineering Informatics* 29 (2015) 196–210.
- [56] S. Dorafshan, R. J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, *Construction and Building Materials* 186 (2018) 1031–1045.
- [57] D. Kiela, E. Grave, A. Joulin, T. Mikolov, Efficient large-scale multi-modal classification, volume 32, 2018.
- [58] W. Wang, D. Tran, M. Feiszli, What makes training multi-modal classification networks hard?, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12695–12705.
- [59] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, *IEEE transactions on pattern analysis and machine intelligence* 41 (2018) 423–443.
- [60] C. Zhang, Z. Yang, X. He, L. Deng, Multimodal intelligence: Representation learning, information fusion, and applications, *IEEE Journal of Selected Topics in Signal Processing* 14 (2020) 478–493.
- [61] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, *Construction and Building Materials* 157 (2017) 322–330.

- [62] K. Zhang, H. Cheng, B. Zhang, Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning, *Journal of Computing in Civil Engineering* 32 (2018) 04018001.
- [63] Z. Fan, Y. Wu, J. Lu, W. Li, Automatic pavement crack detection based on structured prediction with the convolutional neural network, *arXiv preprint arXiv:1802.02208* (2018).
- [64] R. Feldman, B. Rosenfeld, Boosting unsupervised relation extraction by using ner, in: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2006, pp. 473–481.
- [65] M. Atzmueller, P. Kluegl, F. Puppe, Rule-based information extraction for structured data acquisition using textmarker., in: *LWA*, 2008, pp. 1–7.
- [66] A. McCallum, D. Freitag, F. C. Pereira, Maximum entropy markov models for information extraction and segmentation., in: *Icml*, volume 17, 2000, pp. 591–598.
- [67] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, 2001, pp. 282–289.
- [68] Y. Abuzir, . M. O. Abuzir, Constructing the civil engineering thesaurus (cet) using theswb, in: *Computing in Civil Engineering* (2002), 2003, pp. 400–412.
- [69] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, *Journal of machine learning research* 12 (2011) 2493–2537.
- [70] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [71] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

- [72] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for computational Linguistics, 2003, pp. 173–180.
- [73] T. Kudoh, Y. Matsumoto, Use of support vector learning for chunk identification, in: Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop, 2000, pp. 142–144.
- [74] M. Wang, C. D. Manning, Effect of non-linear deep architecture in sequence labeling, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp. 1285–1291.
- [75] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, Y. Shi, Spoken language understanding using long short-term memory neural networks, in: 2014 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2014, pp. 189–194.
- [76] V. Yadav, R. Sharp, S. Bethard, Deep affix features improve neural named entity recognizers, in: Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, 2018, pp. 167–172.
- [77] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv preprint arXiv:1603.01354 (2016).
- [78] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360 (2016).
- [79] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.
- [80] P. Xu, R. Sarikaya, Convolutional neural network based triangular crf for joint intent detection and slot filling, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, 2013, pp. 78–83.

- [81] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, 2013, pp. 6645–6649.
- [82] G. Mesnil, X. He, L. Deng, Y. Bengio, Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding., in: Interspeech, 2013, pp. 3771–3775.
- [83] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, F. Gao, Recurrent conditional random field for language understanding, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 4077–4081.
- [84] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm, in: 9th International Conference on Artificial Neural Networks: ICANN '99, IET, 1999, pp. 850–855.
- [85] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).
- [86] K. M. Tarwani, S. Edem, Survey on recurrent neural network in natural language processing, *Int. J. Eng. Trends Technol* 48 (2017) 301–304.
- [87] W. De Mulder, S. Bethard, M.-F. Moens, A survey on the application of recurrent neural networks to statistical language modeling, *Computer Speech & Language* 30 (2015) 61–98.
- [88] A. Genkin, D. D. Lewis, D. Madigan, Large-scale bayesian logistic regression for text categorization, *Technometrics* 49 (2007) 291–304.
- [89] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, *Journal of Machine Learning Research* 2 (2002) 419–444.
- [90] K. Chen, Z. Zhang, J. Long, H. Zhang, Turning from tf-idf to tf-igm for term weighting in text classification, *Expert Systems with Applications* 66 (2016) 245–260.
- [91] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics* 5 (2017) 135–146.

- [92] C. J. Fillmore, et al., Frame semantics, *Cognitive linguistics: Basic readings* 34 (2006) 373–400.
- [93] Y. Kim, Convolutional neural networks for sentence classification, *arXiv preprint arXiv:1408.5882* (2014).
- [94] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, *arXiv preprint arXiv:1404.2188* (2014).
- [95] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: *Advances in neural information processing systems*, 2015, pp. 649–657.
- [96] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.
- [97] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using bilstm-crf and cnn, *Expert Systems with Applications* 72 (2017) 221–230.
- [98] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [99] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [100] M. Al Qady, A. Kandil, Concept relation extraction from construction documents using natural language processing, *Journal of Construction Engineering and Management* 136 (2009) 294–302.
- [101] N. M. El-Gohary, T. E. El-Diraby, Domain ontology for processes in infrastructure and construction, *Journal of Construction Engineering and Management* 136 (2010) 730–744.
- [102] T. Le, H. D. Jeong, Nlp-based approach to semantic classification of heterogeneous transportation asset data terminology, *Journal of Computing in Civil Engineering* 31 (2017) 04017057.

- [103] C. H. Caldas, L. Soibelman, J. Han, Automated classification of construction project documents, *Journal of Computing in Civil Engineering* 16 (2002) 234–243.
- [104] M. Al Qady, A. Kandil, Automatic clustering of construction project documents based on textual similarity, *Automation in construction* 42 (2014) 36–49.
- [105] N.-W. Chi, K.-Y. Lin, S.-H. Hsieh, Using ontology-based text classification to assist job hazard analysis, *Advanced Engineering Informatics* 28 (2014) 381–394.
- [106] M. Al Qady, A. Kandil, Automatic classification of project documents on the basis of text content, *Journal of Computing in Civil Engineering* 29 (2015) 04014043.
- [107] D. M. Salama, N. M. El-Gohary, Semantic text classification for supporting automated compliance checking in construction, *Journal of Computing in Civil Engineering* 30 (2016) 04014106.
- [108] P. Zhou, N. El-Gohary, Domain-specific hierarchical text classification for supporting automated environmental compliance checking, *Journal of Computing in Civil Engineering* 30 (2016) 04015057.
- [109] F. Zhang, H. Fleyeh, X. Wang, M. Lu, Construction site accident analysis using text mining and natural language processing techniques, *Automation in Construction* 99 (2019) 238–248.
- [110] N. Jung, G. Lee, Automated classification of building information modeling (bim) case studies by bim use based on natural language processing (nlp) and unsupervised learning, *Advanced Engineering Informatics* 41 (2019) 100917.
- [111] D. Wei, B. Wang, G. Lin, D. Liu, Z. Dong, H. Liu, Y. Liu, Research on unstructured text data mining and fault classification based on rnn-lstm with malfunction inspection report, *Energies* 10 (2017) 406.
- [112] N. Kim, S. Hong, Automatic classification of citizen requests for transportation using deep learning: Case study from boston city, *Information Processing & Management* 58 (2020) 102410.
- [113] W. Fang, H. Luo, S. Xu, P. E. Love, Z. Lu, C. Ye, Automated text classification of near-misses from safety reports: An improved deep learning approach, *Advanced Engineering Informatics* 44 (2020) 101060.

- [114] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
- [115] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.
- [116] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [117] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [118] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
- [119] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- [120] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, arXiv preprint arXiv:1411.1792 (2014).
- [121] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2014, pp. 806–813.
- [122] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International conference on machine learning, PMLR, 2014, pp. 647–655.
- [123] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, Gvcnn: Group-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 264–272.
- [124] C. Lin, A. Kumar, Contactless and partial 3d fingerprint recognition using multi-view deep representation, Pattern Recognition 83 (2018) 314–327.

- [125] A. Wang, J. Cai, J. Lu, T.-J. Cham, Mmss: Multi-modal sharable and specific feature learning for rgb-d object recognition, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1125–1133.
- [126] H. Goëau, P. Bonnet, A. Joly, Plant identification in an open-world (lifeclef 2016), in: CLEF: Conference and Labs of the Evaluation Forum, 1609, 2016, pp. 428–439.
- [127] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 945–953.
- [128] A. C. R. Marques, M. M. Raimundo, E. M. B. Cavaleiro, L. FP Salles, C. Lyra, F. J. Von Zuben, Ant genera identification using an ensemble of convolutional neural networks, Plos one 13 (2018) e0192011.
- [129] A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. Van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sánchez, B. Van Ginneken, Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks, IEEE transactions on medical imaging 35 (2016) 1160–1169.
- [130] K. J. Geras, S. Wolfson, Y. Shen, N. Wu, S. Kim, E. Kim, L. Heacock, U. Parikh, L. Moy, K. Cho, High-resolution breast cancer screening with multi-view deep convolutional neural networks, arXiv preprint arXiv:1703.07047 (2017).
- [131] A. Barbosa, T. Marinho, N. Martin, N. Hovakimyan, Multi-stream cnn for spatial resource allocation: A crop management application, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 58–59.
- [132] S. H. Lee, C. S. Chan, P. Remagnino, Multi-organ plant classification based on convolutional and recurrent neural networks, IEEE Transactions on Image Processing 27 (2018) 4287–4301.
- [133] P. Dolata, M. Mrzygłód, J. Reiner, Double-stream convolutional neural networks for machine vision inspection of natural products, Applied Artificial Intelligence 31 (2017) 643–659.

- [134] T.-B. Do, H.-H. Nguyen, H. Vu, T.-L. Le, et al., Plant identification using score-based fusion of multi-organ images, in: 2017 9th International conference on knowledge and systems engineering (KSE), IEEE, 2017, pp. 191–196.
- [135] M. Seeland, P. Mäder, Multi-view classification with convolutional neural networks, *Plos one* 16 (2021) e0245230.
- [136] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.
- [137] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, D. Parikh, Making the v in vqa matter: Elevating the role of image understanding in visual question answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6904–6913.
- [138] R. Zellers, Y. Bisk, A. Farhadi, Y. Choi, From recognition to cognition: Visual commonsense reasoning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6720–6731.
- [139] C. A. Bhatt, M. S. Kankanhalli, Multimedia data mining: state of the art and challenges, *Multimedia Tools and Applications* 51 (2011) 35–76.
- [140] S. K. D'mello, J. Kory, A review and meta-analysis of multimodal affect detection systems, *ACM Computing Surveys (CSUR)* 47 (2015) 1–36.
- [141] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: From unimodal analysis to multimodal fusion, *Information Fusion* 37 (2017) 98–125.
- [142] J. Weston, S. Bengio, N. Usunier, Wsabie: Scaling up to large vocabulary image annotation, 2011.
- [143] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, B. Plank, Automatic description generation from images: A survey of models, datasets, and evaluation measures, *Journal of Artificial Intelligence Research* 55 (2016) 409–442.
- [144] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, D. Parikh, Vqa: Visual question answering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2425–2433.

- [145] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, Multimodal compact bilinear pooling for visual question answering and visual grounding, *arXiv preprint arXiv:1606.01847* (2016).
- [146] A. Lazaridou, E. Bruni, M. Baroni, Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1403–1414.
- [147] M. Baroni, Grounding distributional semantics in the visual world, *Language and Linguistics Compass* 10 (2016) 3–13.
- [148] B. Nojavanasghari, D. Gopinath, J. Koushik, T. Baltrušaitis, L.-P. Morency, Deep multi-modal fusion for persuasiveness prediction, in: *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 284–288.
- [149] H. Wang, A. Meghawati, L.-P. Morency, E. P. Xing, Select-additive learning: Improving generalization in multimodal sentiment analysis, in: *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2017, pp. 949–954.
- [150] V. Vielzeuf, A. Lechervy, S. Pateux, F. Jurie, Centralnet: a multilayer approach for multi-modal fusion, in: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [151] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, R. Fergus, Simple baseline for visual question answering, *arXiv preprint arXiv:1512.02167* (2015).
- [152] E. Shutova, D. Kiela, J. Maillard, Black holes and white rabbits: Metaphor identification with visual features, in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 160–170.
- [153] J.-M. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, F. Jurie, Mfas: Multimodal fusion architecture search, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6966–6975.

- [154] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, *The Journal of Machine Learning Research* 12 (2011) 2211–2268.
- [155] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, *arXiv preprint arXiv:1606.00061* (2016).
- [156] Z. Yang, X. He, J. Gao, L. Deng, A. Smola, Stacked attention networks for image question answering, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.
- [157] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.
- [158] H. Nam, J.-W. Ha, J. Kim, Dual attention networks for multimodal reasoning and matching, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 299–307.
- [159] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al., Oscar: Object-semantics aligned pre-training for vision-language tasks, in: *European Conference on Computer Vision*, Springer, 2020, pp. 121–137.
- [160] H. Tan, M. Bansal, Lxmert: Learning cross-modality encoder representations from transformers, *arXiv preprint arXiv:1908.07490* (2019).
- [161] J. Lu, D. Batra, D. Parikh, S. Lee, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, *arXiv preprint arXiv:1908.02265* (2019).
- [162] T. Li, M. Alipour, D. K. Harris, Context-aware sequence labeling for condition information extraction from historical bridge inspection reports, *Advanced Engineering Informatics* (2021, under review).
- [163] K. Liu, N. El-Gohary, Fusing data extracted from bridge inspection reports for enhanced data-driven bridge deterioration prediction: A hybrid data fusion method, *Journal of Computing in Civil Engineering* 34 (2020) 04020047.

- [164] T. Li, M. Alipour, D. K. Harris, Mapping textual descriptions to condition ratings to assist bridge inspection and condition assessment using hierarchical attention, *Automation in Construction* (2020, under review).
- [165] Z. Sun, P. Tang, Y. Shi, W. Xiong, Visual-semantic alignments for automated interpretation of 3d imagery data of high-pier bridges, in: *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, American Society of Civil Engineers Reston, VA, 2019, pp. 209–216.
- [166] ARTBA, American road and transportation builders association bridge report, 2020. URL: <https://artbabridgereport.org/>.
- [167] S. Saliminejad, N. G. Gharaibeh, Proximity-based outlier detection method for roadway infrastructure condition data, *Journal of Computing in Civil Engineering* 30 (2016) 04015001.
- [168] S. Z. Siabil, N. G. Gharaibeh, Assessing the effect of considering multiple data properties on detecting potential errors in pavement condition data, *Transportation Research Record* 2639 (2017) 39–45.
- [169] P. Chen, R. B. Buchheit, J. H. Garrett Jr, S. McNeil, Web-vacuum: Web-based environment for automated assessment of civil infrastructure data, *Journal of computing in civil engineering* 19 (2005) 137–147.
- [170] T. Li, D. Harris, Automated construction of bridge condition inventory using natural language processing and historical inspection reports, in: *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XIII*, volume 10971, International Society for Optics and Photonics, 2019, p. 109710T.
- [171] T. Li, M. Alipour, D. Harris, Context-aware condition information extraction from bridge inspection reports using bi-directional lstm-crf sequence labeling, *Advanced Engineering Informatics* (2021).
- [172] S. Saliminejad, N. G. Gharaibeh, Impact of error in pavement condition data on the output of network-level pavement management systems, *Transportation research record* 2366 (2013) 110–119.

- [173] L. Gaudette, N. Japkowicz, Evaluation methods for ordinal classification, in: Canadian conference on artificial intelligence, Springer, 2009, pp. 207–210.
- [174] F. H. Administration, National performance management measures; assessing pavement condition for the national highway performance program and bridge condition for the national highway performance program, Federal Register 82 (2017) 14438–14439.
- [175] S. Canny, python-docx: a python library for creating and updating microsoft word (.docx) files, 2019. URL: <https://python-docx.readthedocs.io/en/latest/>.
- [176] K. R. Walus, State of the structures and bridges fiscal year 2018, 2018.
- [177] E. Loper, S. Bird, Nltk: the natural language toolkit, arXiv preprint cs/0205028 (2002).
- [178] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, volume 1, MIT press Cambridge, 2016.
- [179] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014) 1929–1958.
- [180] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [181] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
- [182] N. V. Chawla, Data mining for imbalanced datasets: An overview, in: Data mining and knowledge discovery handbook, Springer, 2009, pp. 875–886.
- [183] S. Visa, A. Ralescu, Issues in mining imbalanced data sets-a review paper, in: Proceedings of the sixteen midwest artificial intelligence and cognitive science conference, volume 2005, sn, 2005, pp. 67–73.

- [184] K. Maeda, S. Takahashi, T. Ogawa, M. Haseyama, Distress classification of class-imbalanced inspection data via correlation-maximizing weighted extreme learning machine, *Advanced Engineering Informatics* 37 (2018) 79–87.
- [185] M. Alipour, D. K. Harris, L. E. Barnes, O. E. Ozbulut, J. Carroll, Load-capacity rating of bridge populations through machine learning: Application of decision trees and random forests, *Journal of Bridge Engineering* 22 (2017) 04017076.
- [186] K. Liu, N. El-Gohary, Learning from class-imbalanced bridge and weather data for supporting bridge deterioration prediction, in: *Advances in Informatics and Computing in Civil and Construction Engineering*, Springer, 2019, pp. 749–756.
- [187] C. Elkan, The foundations of cost-sensitive learning, in: *International joint conference on artificial intelligence*, volume 17, Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.
- [188] I. H. Witten, E. Frank, Data mining: practical machine learning tools and techniques with java implementations, *Acm Sigmod Record* 31 (2002) 76–77.
- [189] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *Journal of artificial intelligence research* 16 (2002) 321–357.
- [190] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.
- [191] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, *Computer-Aided Civil and Infrastructure Engineering* 33 (2018) 731–747.
- [192] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.
- [193] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

- [194] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).
- [195] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, J. Gao, Unified vision-language pre-training for image captioning and vqa, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 13041–13049.
- [196] Python Software Foundation, zipfile — work with zip archives, Accessed: March 28, 2021, <https://docs.python.org/3/library/zipfile.html>, 2021.
- [197] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [198] S. B. Chase, Y. Adu-Gyamfi, A. Aktan, E. Minaie, et al., Synthesis of national and international methodologies used for bridge health indices, Technical Report, United States. Federal Highway Administration, 2016.
- [199] J. Choi, S. J. Dyke, Crowdlim: Crowdsourcing to enable lifecycle infrastructure management, Computers in Industry 115 (2020) 103185.
- [200] D. K. Harris, M. Alipour, S. T. Acton, L. R. Messeri, A. Vaccari, L. E. Barnes, The citizen engineer: Urban infrastructure monitoring via crowd-sourced data analytics, in: Structures Congress 2017, 2017, pp. 495–510.
- [201] M. Alipour, D. K. Harris, A big data analytics strategy for scalable urban infrastructure condition assessment using semi-supervised multi-transform self-training, Civil Structural Health Monitoring (2020).
- [202] U. H. Govindarajan, A. J. Trappey, C. V. Trappey, Intelligent collaborative patent mining using excessive topic generation, Advanced Engineering Informatics 42 (2019) 100955.
- [203] J. Wang, Y.-J. Chen, A novelty detection patent mining approach for analyzing technological opportunities, Advanced Engineering Informatics 42 (2019) 100941.
- [204] P. Kestel, P. Kügler, C. Zirngibl, B. Schleich, S. Wartzack, Ontology-based approach for the provision of simulation knowledge acquired by data and text mining processes, Advanced Engineering Informatics 39 (2019) 292–305.

- [205] D. A. Min, K. H. Hyun, S.-J. Kim, J.-H. Lee, A rule-based servicescape design support system from the design patterns of theme parks, *Advanced Engineering Informatics* 32 (2017) 77–91.
- [206] T. Hartmann, A. Trappey, Advanced engineering informatics-philosophical and methodological foundations with examples from civil and construction engineering, *Developments in the Built Environment* 4 (2020) 100020.
- [207] Z. Yin, Y. Shen, On the dimensionality of word embedding, in: *Advances in Neural Information Processing Systems*, 2018, pp. 887–898.
- [208] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (1997) 1735–1780.
- [209] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, *arXiv preprint arXiv:1508.01991* (2015).
- [210] A. Komninos, S. Manandhar, Dependency based embeddings for sentence classification tasks, in: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1490–1500.
- [211] O. Levy, Y. Goldberg, Dependency-based word embeddings, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.
- [212] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, *Journal of machine learning research* 9 (2008) 2579–2605.
- [213] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [214] N. Reimers, I. Gurevych, Optimal hyperparameters for deep lstm-networks for sequence labeling tasks, *arXiv preprint arXiv:1707.06799* (2017).
- [215] Y. Bengio, P. Simard, P. Frasconi, et al., Learning long-term dependencies with gradient descent is difficult, *IEEE transactions on neural networks* 5 (1994) 157–166.

- [216] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International conference on machine learning, 2013, pp. 1310–1318.
- [217] R. Caruana, S. Lawrence, C. L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: Advances in neural information processing systems, 2001, pp. 402–408.
- [218] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of machine learning research* 2 (2001) 265–292.
- [219] E. Agichtein, L. Gravano, Snowball: Extracting relations from large plain-text collections, in: Proceedings of the fifth ACM conference on Digital libraries, ACM, 2000, pp. 85–94.
- [220] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, in: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Association for Computational Linguistics, 2004, p. 22.
- [221] K. Fundel, R. Küffner, R. Zimmer, Relex—relation extraction using dependency parse trees, *Bioinformatics* 23 (2006) 365–371.