Automating the Generation of Construction Checklists

Pay Item

ID	Description	Unit	Type
201-02015	CLEARING AND GRUBBING	LS	S

Standard Specification

Inspection Form

[Pay Item] 201-02015 CLEARING AND GRUBBING

ID	Check Question
201-004	Surface objects, trees, stumps, roots, and other protruding obstructions not designated to remain are cleared, grubbed, and removed from within the limits of the work?
201-005	Undisturbed sound stumps, roots, and non-perishable solid objects do not extend more than 4 in. above the ground line or low water level?
201-007	Except in areas to be excavated, stump holes and other holes from which obstructions are removed are backfilled with suitable material and compacted in accordance with 203.23?

**Hubo Cai, JungHo Jeon, Xin Xu,
Yuxi Zhang, Liu Yang**

AUTHORS

Hubo Cai, PhD, PE
Professor of Civil Engineering
Lyles School of Civil Engineering
(765) 494-5028
hubocai@purdue.edu
Corresponding Author

JungHo Jeon
Xin Xu
Yuxi Zhang
Liu Yang
Graduate Research Assistants
Lyles School of Civil Engineering
Purdue University

JOINT TRANSPORTATION RESEARCH PROGRAM

The Joint Transportation Research Program serves as a vehicle for INDOT collaboration with higher education institutions and industry in Indiana to facilitate innovation that results in continuous improvement in the planning, design, construction, operation, management and economic efficiency of the Indiana transportation infrastructure. https://engineering.purdue.edu/JTRP/index_html

Published reports of the Joint Transportation Research Program are available at http://docs.lib.purdue.edu/jtrp/.

NOTICE

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views and policies of the Indiana Department of Transportation or the Federal Highway Administration. The report does not constitute a standard, specification or regulation.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA/IN/JTRP-2020/23	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Automating the Generation of Construction Checklists		**5. Report Date** October 2020
		6. Performing Organization Code
7. Author(s) Hubo Cai, JungHo Jeon, Xin Xu, Yuxi Zhang, and Liu Yang		**8. Performing Organization Report No.** FHWA/IN/JTRP-2020/23
9. Performing Organization Name and Address Joint Transportation Research Program Hall for Discovery and Learning Research (DLR), Suite 204 207 S. Martin Jischke Drive West Lafayette, IN 47907		**10. Work Unit No.**
		11. Contract or Grant No. SPR-4422
12. Sponsoring Agency Name and Address Indiana Department of Transportation (SPR) State Office Building 100 North Senate Avenue Indianapolis, IN 46204		**13. Type of Report and Period Covered** Final Report
		14. Sponsoring Agency Code

15. Supplementary Notes
Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract
Construction inspection is a critical component of INDOT's quality assurance (QA) program. Upon receiving an inspection notice/assignment, INDOT inspectors review the plans and specifications to identify the construction quality requirements and conduct their inspections accordingly. This manual approach to gathering inspection requirements from textual documents is time-consuming, subjective, and error-prone. This project addresses this critical issue by developing an inspection requirements database along with a set of tools to automatically gather the inspection requirements and provide field crews with customized construction checklists during the inspection with the specifics of what to check, when to check, and how to check, as well as the risks and the actions to take when noncompliance is encountered. This newly developed toolset eliminates the manual effort required to acquire construction requirements, which will enhance the efficiency of the construction inspection process at INDOT. It also enables the incorporation of field-collected data to automate future compliance checking and facilitate construction documentation.

17. Key Words		18. Distribution Statement
quality assurance, construction inspection, risk assessment, knowledge management, ontology, digital inspection system, construction inspection, automatic and dynamic checklist, natural language processing		No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified	21. No. of Pages 29	22. Price

Form DOT F 1700.7 (8-72)   Reproduction of completed page authorized

EXECUTIVE SUMMARY

Introduction

Construction inspection is a critical component of the quality assurance/quality control (QA/QC) program at the Indiana Department of Transportation (INDOT). In the current practice, inspectors manually gather the construction requirements for a construction activity from the plans and specifications, which is time-consuming and error-prone, and heavily relies on the experience of the inspectors and their subjective interpretation of the textual requirements. Automating the requirements-gathering process and presenting it in a consistent and easy-to-use format can greatly enhance inspection efficiency and accuracy and, consequently, address the resource shortage at INDOT.

The objective of this project is to develop a tool to automate the generation of a customized and dynamic checklist that contains only the applicable, specific information for a construction activity at the pay item level with adequate details to guide the construction inspection process. Besides basic information, the resulting checklist (composed of a list of check items) includes the acceptance criteria, the inspection frequency, and the associated risk. The checklist also has mechanisms that connect individual checklist items to relevant, available training materials that come in varying formats, such as photos, videos, textual documents, and websites. The project scope includes Divisions 200, 300, 400, 500, and 700.

A relational database was designed and constructed in this project to store the pay items and check items as well as the construction process and stages. A graphical user interface and specific tools were developed to perform the following functions: (1) automatic generation of a customized checklist at the pay item level; (2) access to a checklist display aligned with the repetitive/cyclical nature of construction workflows; (3) navigation between cross-referenced check items; (4) subgrouping options based on the responsibility, risk level, and inspection frequency; and (5) linking options to accommodate photos, videos, textual documents, and websites. Natural language processing (NLP) and machine learning algorithms were explored to extract the inspection requirements from textual documents and restructure them into the checklist questions.

Findings

The main findings of this project are as follows:

- The database approach (including both the database design and the user interface) is capable of achieving the desired functionalities to automatically generate customized checklists at the pay item level.
- The division-section-subsection-requirement sentences hierarchy can help organize the construction requirements and the applicable pay items. In particular, subsections play a central role in linking pay items, construction processes/stages, and check items.
- Testing the four mechanisms that could accommodate the inspection documents and training materials in varying formats revealed the following:

  - A URL link is sufficient to connect checklist items with websites.
  - Embedding files in the database is impracticable due to size constraints and data duplication.
  - Organizing files under folders and linking pay items to folders are efficient approaches but they require additional effort to explore and navigate through the file system.
  - The variant version, wherein each check item is linked to a central linking file that contains links to all the applicable documents stored on the cloud, will save storage space and eliminate data redundancy. This mechanism is recommended to INDOT for implementation.

- NLP techniques developed for the automatic extraction of the check items from the *INDOT Standard Specification 2018* achieves an approximate accuracy of 90%.

  - A regular expression (RE) is efficient in identifying the hierarchy (subsections-section-division) of a file and separating sentences based on their textual patterns.
  - The sentence classifier composed of GloVe (a word embedding algorithm) and a convolutional neural network (CNN) (a deep learning algorithm) results in the highest accuracy and the least loss.
  - Using training samples from all divisions, a higher percentage of data, and/or a validation set improves the classification performance.

- A valuable dataset that is comprised of 3,171 sentences extracted from the *INDOT Standard Specification 2018* and labeled with the correct classes ("1" for requirement and "0" for nonrequirement) can be used to train models and automatically extract inspection requirements from other versions of INDOT's specifications to address the specification modifications issue.

Implementation

The main deliverables of this project are the relational inspection database and NLP algorithm. There are at least three options for implementing the database.

1. The database can be directly used as a stand-alone system. A user can import the pay items from the INDOT contract information book (CIB) into the database and use the already-developed user interface to generate a customized checklist for the chosen pay item(s) and exporting it in varying formats (Excel, Word, PDF).
2. The database can be connected to the field inspection and documentation tool being developed. When an inspector chooses a specific pay item, the applicable construction requirements can be extracted from the database. The inspection results are then saved and stored together with the quantity check.
3. The database can be imported into the enterprise database at INDOT and used either as a standalone application or connected to the field inspection and construction documentation system.

The NLP algorithm can achieve about 90% accuracy. The classification model was trained based on INDOT's Standard Specification 2018 but can be directly used on other versions to extract construction requirements automatically. Users are cautioned that the scope of this project included Divisions 200, 300, 400, 500, and 700, and thus only applies to pay items and construction requirements in those divisions.

At the time of this writing, a team composed of experts from the Management Information System (MIS) unit and the Construction group at INDOT is working on the system implementation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Background

Construction inspection is a critical component of a quality assurance/quality control (QA/QC) program to ensure the quality and long-term performance of constructed infrastructure. The current practice at the Indiana Department of Transportation (INDOT) requires construction engineers and inspectors to conduct an intensive inspection and documentation program based on INDOT's *Standard Specification 2018* and *General Instructions to Field Employees Manual.* Due to the large number of construction projects statewide, the increasing complexity of modern infrastructure projects, and the retirements and departures to the private sector of experienced and knowledgeable employees, INDOT faces a huge challenge as field crews are extremely overloaded (Xu et al., 2019a). Therefore, a critical need exists for an effective inspection toolset to help improve the efficiency of inspectors. Towards that end, this study aimed to develop a toolset to provide inspectors with customized construction checklists that are appropriate for the work under inspection with all the necessary details regarding what to check, when to check, and how to check as well as the risks and actions when noncompliance is encountered. This toolset is expected to greatly reduce the workload for inspectors and enhance the effectiveness of the construction inspection process by enabling advanced planning and providing educational materials and inspection guidance in real-time.

1.2 Problem Statement

In the previous JTRP Project No. SPR-4002, checklists were developed and organized by the specification sections covering earthwork, PCCP and HMA pavements, and bridge decks, which were the starting point for the current project (Xu et al., 2019b). The resulting inspection forms were tested in the field in the manner of a mock-up drill after construction. While the newly developed checklists have proven their effectiveness in reducing the workload and enhancing productivity, a few areas were identified as critical limitations that must be addressed in order to make the revised checklists a powerful tool for field inspectors and engineers. The main issue is that the checklists and forms are static in nature. They are organized by specification sections; and when inspecting a specific pay item, an inspector must be able to identify all the relevant specification sections and then use those sections' associated checklists. Therefore, the inspector has to manually identify all the relevant sections and use multiple checklists for a single pay item. Furthermore, these forms contain both applicable and nonapplicable check items, which may cause confusion and is counter-productive. The relevant limitations are as follows:

- Lack of cross-referencing capability (i.e., the specification sections are often cross-referenced, and it is very common that several sections all contain relevant construction requirements of a specific pay item).

- Lack of flexibility in grouping check items in various ways, such as the responsibility.
- Lack of necessary inspection details, such as the acceptance criteria, the consequence of failure or ignorance thereof, and the necessary training materials.
- Lack of capability to accommodate photos, images, and even short videos.
- Lack of an automated process to develop inspection checklists.

1.3 Objective

The goal of this project was to maximize the practical value of the checklist approach to INDOT construction inspection by automating the generation of customized checklists at the pay item level. The objective of this project was to develop a tool to automate the generation of a customized and dynamic checklist that contains applicable (only), specific information with adequate details to guide the construction inspection process. Besides basic information, the resulting checklist should include the acceptance criteria, inspection frequency, associated risk (consequence of failure and its impact to INDOT), and videos and photos of available training and guidance materials to assist construction inspectors. The project scope was set to include Divisions 200, 300, 400, 500, and 700. The individual tasks to achieve the objective were as follows.

1. Automate the generation of customized checklists at the pay item level by developing a maintainable database to store the check items and link them to applicable pay items within a user-friendly interface to generate a customized checklist for specific pay items that contains the applicable check requirements/items only. Develop a business process as well to update the database when there are changes/modifications to the specifications.

2. Create flexibility for accommodating pay items that are associated with repetitive/cyclical workflows. For example, the work cycle in piping involves trenching-installation-backfill-compaction. The checklist thus can be lengthy, but not all of the items are necessary at a specific stage. For instance, when checking trenching, the inspector does not need to see the items under installation, backfill, or compaction. For this group of pay items, the link between the check items and the pay items needs to be further detailed down to the individual stages of the workflow. Also needed are mechanisms such as a collapse/expand dropdown and allowing inspectors to see and use check items when needed, rather than filling the computer screen with all check items for the work, many of which are not necessary at a specific stage.

3. Provide a function to display the check items and inspection results in cross-referenced sections to eliminate duplicate checking efforts and provide inspectors with data (previous checking results). The specifications initially need to be examined in detail to identify all the cross-referencing possibilities and then to implement them as links in the database so that they show up as hyperlinks for inspectors to navigate.

4. Incorporate additional customization and subgrouping functions in the database. Information such as

responsibility, acceptance criteria, and why an item needs to be inspected or the consequences if not inspected should be included in the database. A subgrouping based on user-specified criteria also should be included by creating additional columns in the database tables, setting up the links, and developing a user interface for easy data input.

5. Provide accommodations for photos and videos of inspection documentation and training materials. Inspectors take photos, and training materials use photos and videos. The objective is to allow inspectors to upload inspection photos and videos and to access training videos and guidance materials when needed. The ultimate goal is to link to training materials and construction guidelines in the form of photos, narratives, videos, or any combination of these three formats to provide necessary (re)training to both novice and experienced inspectors on an as-needed basis.

6. Design new tools to automatically extract check items from text documents for selective pay items or sections from 600, 800, and RSPs. This new automated compliance checking method should compare the inspection results to the acceptance criteria and automatically fill out the inspection form with results of Pass/Fail and should provide the actions to take when encountering a Fail to provide guidance to inspectors.

The main deliverables of this project were an intelligent database of checklists with user-friendly tools. The resulting database is capable of automatically generating customized inspection checklists with adequate details for any given pay item. It provides a powerful tool to help construction inspectors conduct construction inspections faster and more efficiently.

2. AUTOMATIC GENERATION OF CUSTOMIZED CHECKLIST AT THE PAY ITEM LEVEL

2.1 Introduction

The focus of this task was designing and constructing a maintainable database to store check items and linking them to applicable pay items in specific subsections (Figure 2.1)

A method to retrieve pay items from a project's CIB also was devised and implemented as a part of the database with the following workflow. First, the project-specific pay items are retrieved and uploaded into the database to formulate a searchable pool of pay items. Second, upon selection of a pay item, the linking mechanism is initiated to retrieve the applicable check items with the details covering the check frequency, acceptance criteria, risk level, and training and guidance resources. The user can use the retrieved checklist directly or export it to Word, Excel, or PDF formats.

2.2 Linking Check Items to Pay Items

The check items were extracted from *INDOT's Standard Specification 2018*. Figure 2.2 illustrates how the check items were extracted. Each check item has an ID, check question, item type (check or information), and referenced subsections. The specification lists the applicable pay items for each section; and by examining each pay item in detail, their applicable subsections were identified. Table 2.1 illustrates the pay items in *Section 201* and their corresponding subsections. For example, *pay item 201-01025 Scalping* corresponds to *subsection 201.04*. Therefore, by using the subsections as the junction, the pay items and their applicable check items are linked to each other.

Figure 2.3 illustrates the database design for generating the customized checklist of check items at the pay item level. Table *tblPayItem* contains the essential attributes of the pay item; and Table *tblCheckItem* contains all the check details. Table *tblPayItemToSubsection* joins *tblPayItem* via the [PayItemID] attribute and *tblCheckItem* by matching the [SubsectionNumber] and [Subsection] attributes. Tables *tblSubsectioninfo* and *tblSectionInfo* provide the numbering and title information and, when needed, can be joined to other tables based on the numbering information. Note that Tables *tblSubsectioninfo* and *tblSectioninfo* can be joined via the [Section] attribute (arrow not shown in the figure). The sample SQL script below generates the customized checklist for a pay item with ID XXX-XXXXX.

SELECT * FROM (((tblPayItem INNER JOIN tblPayItemToSubsection ON tblPayItem.PayItemID = "XXX-XXXXX) INNER JOIN tblSectionInfo ON tblPayItem.SectionNumber = tblSectionInfo.SectionNumber) INNER JOIN tblSubsectionInfo ON tblPayItemToSubsection.SubsectionNumber = tblSubsectionInfo.SubsectionNumber) INNER JOIN tblCheckItem ON tblPayItemToSubsection.SubsectionNumber = tblCheckItem.Subsection.

2.3 Extract Pay Items from CIB

The INDOT CIB contains a list of the pay items for a specific project. Each pay item is associated with the description, the quantity and units, the unit price, and the bid amount. Figure 2.4 illustrates the procedure to extract the pay items and import them into the database. Starting from the CIB in PDF format, an Excel spreadsheet is populated with the list of pay items and then uploaded into the database as a new table.

Figure 2.1 Concept of linking mechanism.

Figure 2.2 Extraction of check items from specification.

TABLE 2.1
Applicable Pay Items Under Section 201

Pay Item ID	Description	Subsection	Subsection Title
201-01015	CLEARING AND GRUBBING	201.03	Clearing and Grubbing
201-01025	SCALPING	201.04	Scalping
201-02245	TREE 6 IN., REMOVE	201.05	Hedge Removal
201-06587	CLEARING AND GRUBBING	201.03	Clearing and Grubbing
201-12044	CLEARING AND GRUBBING	201.03	Clearing and Grubbing
201-52370	CLEARING RIGHT OF WAY	201.03	Clearing and Grubbing
201-90788	DEBRIS, REMOVE STRUCTURE NO.	201.03	Clearing and Grubbing

Figure 2.3 Database design for generating a customized checklist at the pay item level.

Figure 2.4 Extraction of pay items from CIB.

Figure 2.5 Sample illustration: Generating checklist at the pay item level.
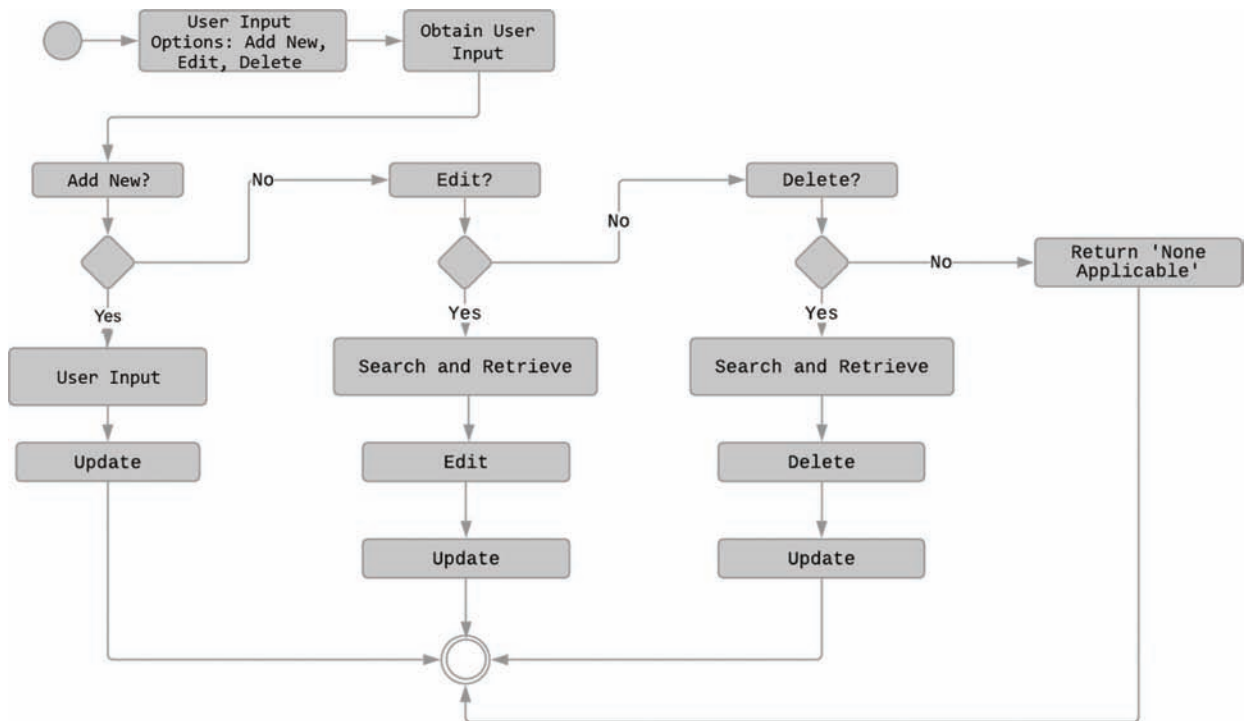
Figure 2.6 Business process to update database when specification changes.

2.4 Implementation and Illustration

A corresponding user interface was designed and implemented as part of the digital inspection system using MS Access 2016. Figure 2.5 illustrates how it works with examples. All the project-specific pay items retrieved from the CIB are shown in the dropdown list (A). When the user chooses a pay item (e.g., *pay item 201-01015*), the applicable check items (e.g., *Check Item ID 201-004, 005, 007, 008, and 010*) with related information (e.g., inspection frequency, type, and responsibility) are retrieved and displayed in a subform (B). The user can view the checklist directly in the database or export it to Word, Excel, or PDF formats (C). Note the connection is established via *subsection 201.03*.

2.5 Business Process to Update Database

A business process (Figure 2.6) was designed to update the database when there are changes or modifications to the specification. This process aligns with the linking mechanism discussed in the previous section. Depending on the changes in the specification, three options (adding new, edit, and delete) can be taken.

3. PROVIDING FLEXIBILITY FOR ACCOMODATING PAY ITEMS THAT ARE REPETITIVE WORKFLOWS

3.1 Introduction

This task focused on pay items of a repetitive/cyclical nature. Many pay items involve repetitive workflows;

for example, piping follows a four-step process: trenching, installation, backfill, and compaction. As the piping activity progresses, this process is repeated. Correspondingly, applicable check items can be grouped into these four stages and retrieved and displayed for the current stage under inspection. For example, when checking construction work at the trenching step, only the check items categorized under this step will be retrieved as the inspector does not need check items for the installation, backfill, or compaction stages.

To accomplish the objective in this task, three steps were taken: (1) identifying the pay items that have repetitive/cyclical workflows, (2) determining the repetitive steps/stages, and (3) designing the mechanism to group the applicable check items into corresponding steps/stages.

3.2 Identifying All Pay Items that are Repetitive/Cyclical in Nature

All 2,220 pay items in Divisions 200, 300, 400, 500, and 700 were examined, and 1,116 pay items were identified as having repetitive cycles. Table 3.1 shows the breakdown of the pay items examined and the pay items identified at the division level. On average, 50% of the pay items involved repetitive workflows. Figure 3.1 illustrates that all the SEAL COAT pay items shared the same repetitive cycle, which was composed of (1) preparation of surface, (2) applying asphalt material, (3) applying cover aggregate, (4) rolling operation, and (5) sweeping operation.

3.3 Determining Repetitive Steps

Similar pay items were grouped based on the construction process they share. A total of 29 construction processes were identified (Table 3.2). For example, the construction process of PCCP PATCHING is a three-stage process composed of (1) PCCP removal, (2) concrete mixing and transportation, and (3) concrete placement. These construction processes then were linked to their corresponding pay items.

3.4 Grouping Applicable Check Items to Corresponding Stages

The focus in this task was to devise a mechanism to group the applicable check items to their corresponding construction stages. Figure 3.2 illustrates that all the construction stages are assigned to the applicable subsections, and the check items referencing the same subsections are linked (e.g., *CheckItemID 404-014 and 404-015*) to Stage 4–rolling operation in the SEAL COAT construction process.

3.5 Implementation and Illustration

Figure 3.3 illustrates the database design for accommodating pay items that have repetitive/cyclical workflows, where the arrows illustrate the linkage among the pay items, construction processes, construction stages, and check items. Table *tblConstructionProcess* contains two attributes, [PayItemID] and [ConstructionProcess], which associate a construction process to its corresponding pay items. Table *tblConstructionStage* lists all the stages (with stage numbering) involved in a specific process. Table *tblStageToSubsection* includes information about the association between the construction stages and the subsections. Since *tblCheckItem* has the [Subsection] attribute that stores the numbering of the source subsection, joining all these tables using the foreign key-primary key constraints (indicated by arrows) organizes the applicable check items to their corresponding construction stages. This was accomplished by executing the following SQL script:

SELECT*FROM (tblPayItem INNER JOIN ((tbl ConstructionProcess INNER JOIN tblConstruction Stage ON tblConstructionProcess.ConstructionProcess = tblConstructionStage.[ConstructionProcess]) INNER JOIN tblStageToSubsection ON tblConstructionStage. ConstructionStage = tblStageToSubsection.Construction Stage) ON tblPayItem.PayItemID = tblConstruct ionProcess.PayItemID) INNER JOIN tblCheckItem ON tblStageToSubsection.Subsection = tblCheckItem.Subsection.

A user interface was designed and implemented as part of the digital inspection system. Figure 3.4

TABLE 3.1
Results of Identified Pay Items that are of a Repetitive Nature

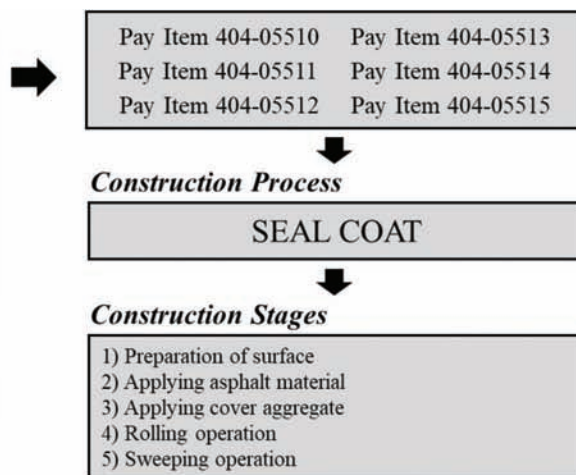| Division	Number of Pay Items	Identified Pay Items of Repetitive Nature	Percentage (%)
200	234	22	9.4
300	71	20	28.2
400	131	120	91.6
500	63	60	95.2
700	1,721	894	51.9
Total	*2,220*	*1,116*	*50.3 (avg)*

Figure 3.1   Process of repetitive pay item identification with corresponding process and stages.

TABLE 3.2
Identified Construction Processes

Division	Construction Process
200	Subgrade treatment (2), Geogrid, Geotextile for pavement (2), Chemical modification of soils (7), Drying soils (6)
300	Aggregate compaction (3), Subbase (3), Aggregate (3), PCC base (3)
400	CMA (5), HMA (4), Warranted micro-surfacing (2), QC/QA-HMA (4), Seal coat (5), Ultrathin bonded wearing (4)
500	PCCP (8), QC/QA-PCCP (10), PCCP patching (3)
700	Structural concrete (17), Reinforcing bars (3), Concrete floor slabs (5), Mortar (4), Structural steel (3), Pipes (5), Structural plate pipe (3), Bridge deck overlays (9), Manhole/Inlet/Catch basins (2), Slip lining of existing pipe (4), Trenchless pipe (11)

Note: The numbers in brackets represent the number of stages involved in the identified construction process.
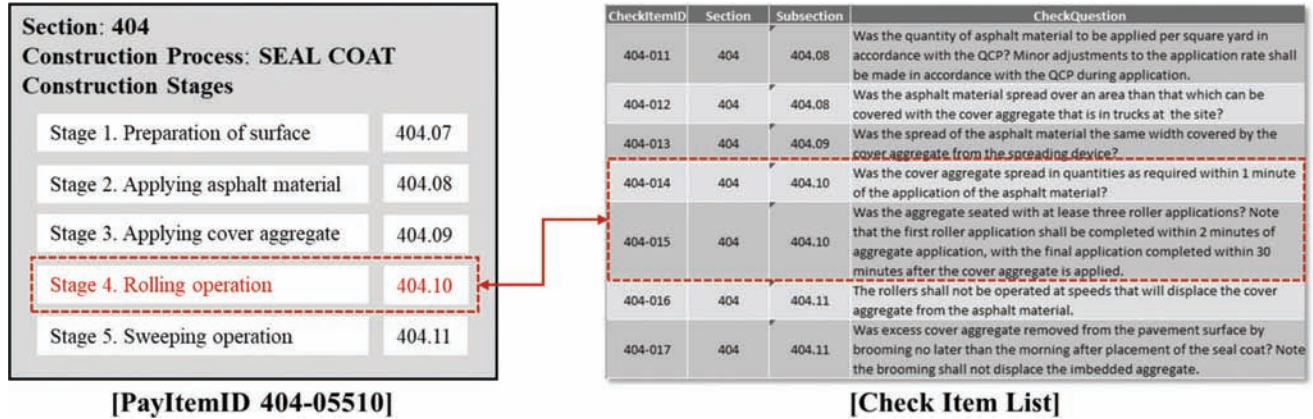

Figure 3.2 Grouping check items to the corresponding stage.
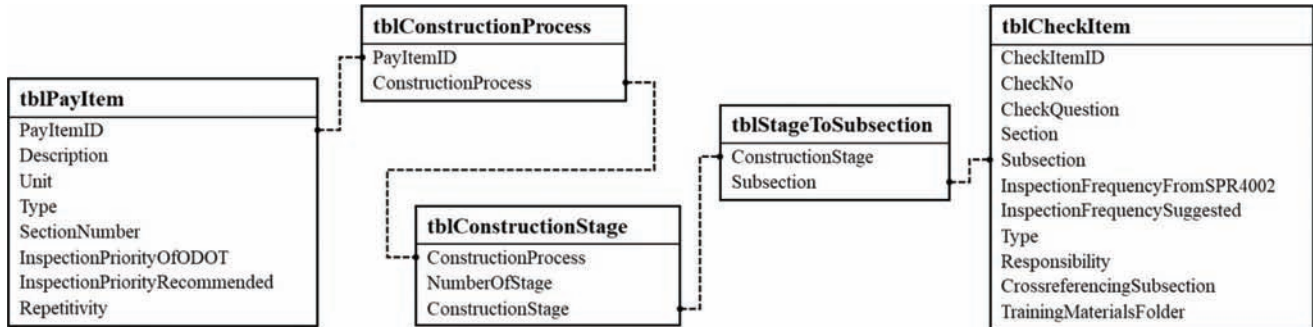

Figure 3.3 Database design for accommodating pay items of a repetitive nature.

illustrates the interface with an example. The dropdown list (A in Figure 3.4) lists the pay items that are repetitive in nature. For a selected pay item, its corresponding construction process and its composing stages are retrieved and displayed in a subform (B in Figure 3.4). As a user toggles through these stages, the corresponding check items are displayed in another subform (C in Figure 3.4).

Figure 3.4 Sample illustration: Accommodating pay items of a repetitive nature.

4. CROSS-REFERENCING

4.1 Introduction

This task focused on enabling the cross-referencing of the relevant requirements that are captured as separate check items. It is common in the specification for a requirement to refer to relevant requirements in other sections or subsections, as shown in Figure 4.1. For example, *CheckItemID 201-007* cross-references *subsection 203.23*. Among all 2,415 check items, 304 check items cross-referenced other subsections. This function was critically needed to display cross-referenced information to construction inspectors to eliminate duplicate checking efforts.

4.2 Database Design and Implementation

Figure 4.2 (left) illustrates the database design for cross-referencing. A new column, [Crossreferencing Subsection], was added to table *tblCheckItem*. This new column was populated with the cross-referenced subsection numbers. Based on the matching of [CrossreferencingSubsection], which is the subsection cross-referenced by the current check item, and [Subsection], which is the subsection to which a check item belongs, the applicable check items under the corresponding subsection can be retrieved and displayed using the SQL script below. The working mechanism is illustrated on the right side of Figure 4.2.

CheckItemID	CheckQuestion	Section
201-007	Except in areas to be excavated, stump holes and other holes from which obstructions are removed are backfilled with suitable material and compacted in accordance with 203.23?	201
201-008	Low hanging, unsound, dead, or unsightly branches are removed and branches of trees extending over the roadbed are trimmed to give a clear height of 20 ft above the roadbed?	201
201-009	Tree wound dressing required for cut or scarred surfaces of trees or shrubs selected for retention is in accordance with 914.09(c)?	201
201-010	All materials removed or excavated were either reused in the work; or recycled, burned, or disposed according to 203.08 and applicable laws, ordinances, rules, and regulations?	201
201-011	All damage is repaired or replaced in accordance with 621.11?	201
201-012	Hedges and shrubs are pulled or grubbed?	201

Figure 4.1   Examples of check items that cross-reference other sections.

tblCheckItem [Database Design]
- CheckItemID
- CheckNo
- CheckQuestion
- Section
- Subsection
- InspectionFrequencyFromSPR4002
- InspectionFrequencySuggested
- Type
- Responsibility
- CrossreferencingSubsection
- TrainingMaterialsFolder

tblCheckItem [A] [SQL Statement Illustration]
- CheckItemID
- CheckNo
- CheckQuestion
- Section
- Subsection
- InspectionFrequencyFromSPR4002
- InspectionFrequencySuggested
- Type
- Responsibility
- CrossreferencingSubsection
- TrainingMaterialsFolder

tblCheckItem [B]
- CheckItemID
- CheckNo
- CheckQuestion
- Section
- Subsection
- InspectionFrequencyFromSPR4002
- InspectionFrequencySuggested
- Type
- Responsibility
- CrossreferencingSubsection
- TrainingMaterialsFolder

Figure 4.2   Database design and SQL statement for cross-referencing.

*SELECT * FROM tblCheckItem as A, tblCheckItem as B WHERE ([A].[CrossreferencingSubsection] = [B].[Subsection]);*

A user interface was developed to display the check items in the cross-referenced subsection. Figure 4.3 illustrates the interface using *PayItemID 201-01015*

CLEARING AND GRUBBING as an example. The applicable check items for this pay item are displayed in B in Figure 4.3. Check item 201-007 cross-references subsection 203.23. All the check items under subsection 203.23 are displayed in C in Figure 4.3.

Figure 4.3 Sample illustration: Cross-referencing.

5. DEVELOPING FURTHER CUSTOMIZATION AND SUBGROUPING CAPABILITY

5.1 Introduction

The objective of this task was to further enhance the customization capability in the tool by incorporating information (e.g., inspection frequency, responsibility, and type) and enabling a subgrouping based on user-specified criteria. For example, based on the responsibility (e.g., PE/S, design reviewer, or inspector), a group of corresponding check items can be displayed to the user. To achieve this functionality, several tools (e.g., combo box, filtering, and events builder) provided by MS Access and Visual Basic for Application (VBA) code were used. The resulting system is expected to provide project engineers with the flexibility to customize the retrieved check list.

5.2 Implementation and Illustration

A corresponding user interface was designed and implemented to enable further customization and subgrouping. Figure 5.1 illustrates the interface, using *PayItemID 201-52370 CLEARING RIGHT OF WAY* as an example. The applicable check items are retrieved and displayed with details (e.g., inspection frequency, type of check item, and responsibility) (B in Figure 5.1). Users can filter the results based on responsibility (role) (C in Figure 5.1) or inspection frequency (D in Figure 5.1). Figure 5.2 shows the filtered results when "Inspector" is chosen. Figure 5.3 illustrates the results

Figure 5.1 Sample illustration: Further customization and subgrouping.

Figure 5.2 Filtered results based on the responsibility role.

Figure 5.3 Filtered results based on the inspection frequency.

for all the check items with an inspection frequency of 3, which refers to end product inspection.

6. ACCOMMODATION FOR PHOTOS AND VIDEOS OF INSPECTION DOCUMENTATION AND TRAINING MATERIALS

6.1 Introduction

The objective of this task was to develop a mechanism to accommodate photos, videos, and Word and PDF files for construction inspectors to access training/guidance materials when needed. Depending on the original source and format of the materials, the following three mechanisms were explored and assessed:

1. URL link to external websites
2. Embedding files in the database
3. Organizing materials in folders and linking to:

    3.1 Individual files
    3.2 Explorable folders

6.2 Mechanism 1: URL Link to External Websites

This mechanism established a connection between a check item and relevant website (both INDOT and non-INDOT) materials. Figure 6.1 illustrates the working mechanism. A column entitled "Link" was added to table *tblCheckItems* to store the URL (B in Figure 6.1). These URL links are clickable (i.e., a click on the URL link opens the linked website in a web browser) (C in Figure 6.1).

The main findings were as follows:

- Minimum local storage is required since the materials are stored in an external repository.
- URL links must be entered by the user, which is tedious.
- The materials cannot be controlled by the user.
- It is challenging to maintain concurrent correctness of both the links and the web content.
- A Wi-Fi connection is necessary to access the materials.

6.3 Mechanism 2: Embedding Files in the Database

This mechanism associates the check items to files in varying formats (e.g., images/photos, videos, PDF, Word, Excel) by embedding the files in the database. MS Access provides an *Attachment* feature that allows users to store multiple files of varying formats in a single field. A new column, [Attachment], was added to the table *tblCheckItem* (Figure 6.2). This new column is populated with the file information associated with each check item. In this way, the files are embedded in their native formats in the database.

Figure 6.3 shows a demo for this mechanism. The selected *check item 201-004* was associated with four files. A in Figure 6.3 shows the count of associated files, and B in Figure 6.3 provides preview information for all the associated files. Clicking the A button activates a popup window (C) where the user can manage the associated files (add and remove).

The main findings were as follows:

- This mechanism is very efficient in retrieving associated files.
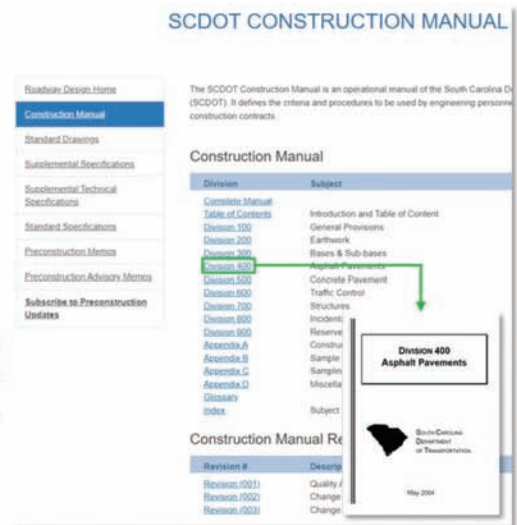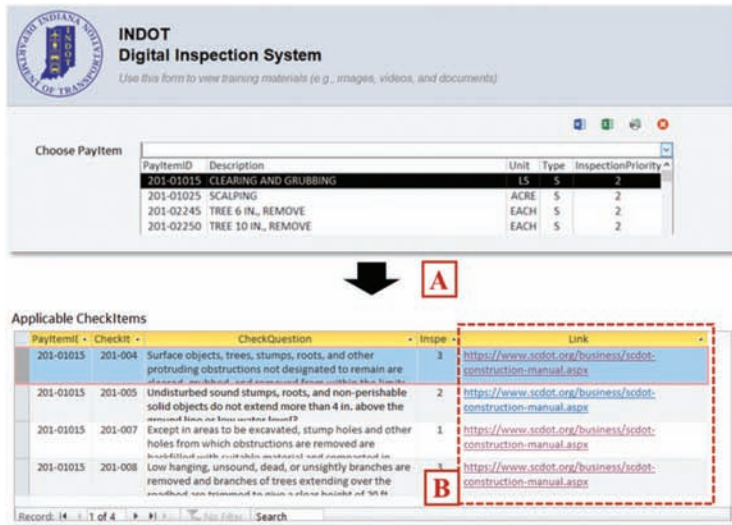
Figure 6.1 Sample illustration: URL link to external website.
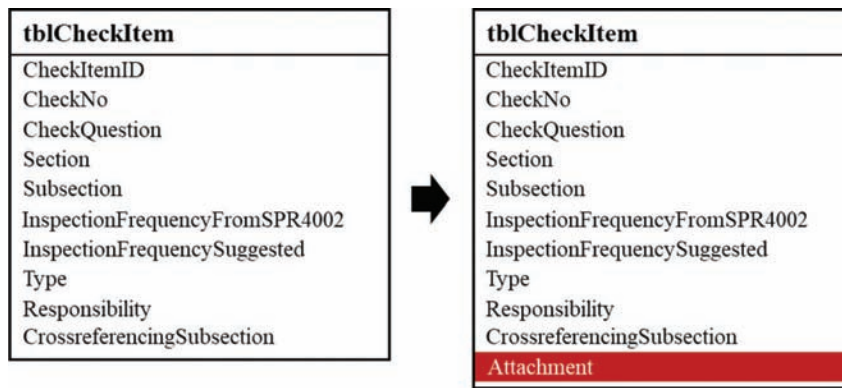
Figure 6.2 Attachment feature in access.

- Managing the associated files is tedious and time-consuming.
- The size limit is the main concern. Assuming a typical image size of 2.5 megabytes, it would only take 750 photos to reach the maximum capacity of 2 GB in an MS Access database. Enterprise databases are less restrictive on the size capacity; however, considering the thousands of check items associated with tens of photos, videos, and PDF files, the total size can reach 100 GB easily.

6.4 Mechanisms 3.1 and 3.2: Organizing Materials in Folders and Linking to Individual Files or Explorable Folders

Two mechanisms were devised and tested as alternatives to Mechanism 2 to establish the connection between the check items and the files. Mechanism 3.1 connects a check item to its associated files separately. Figure 6.4 illustrates that multiple links may have to be established for a check item, and the number of links depends on the number of associated files.

Mechanism 3.2 connects a check item to a folder that contains all the associated files. Figure 6.5 shows that only one link to the folder needs to be established for a check item.

Corresponding user interfaces were developed to test the mechanisms. For Mechanism 3.1, Figure 6.6 shows multiple links in A, each of which corresponds to a file in B. Clicking a thumbnail in B launches the corresponding file in its default application (C).

Figure 6.7 illustrates a demo for Mechanism 3.2. Each check item is associated with a link (in A) to an explorable folder. Clicking the folder link in A activates the file window (in B), which displays thumbnails for each of the files in the folder. Users can click on any of these thumbnails to display the selected file (C).

The main findings were as follows:

- For Mechanism 3.1, the main challenge was the data structure. Different check items can have a different number of files associated with them. An additional table may be needed with numerous rows of check item-file pairs.
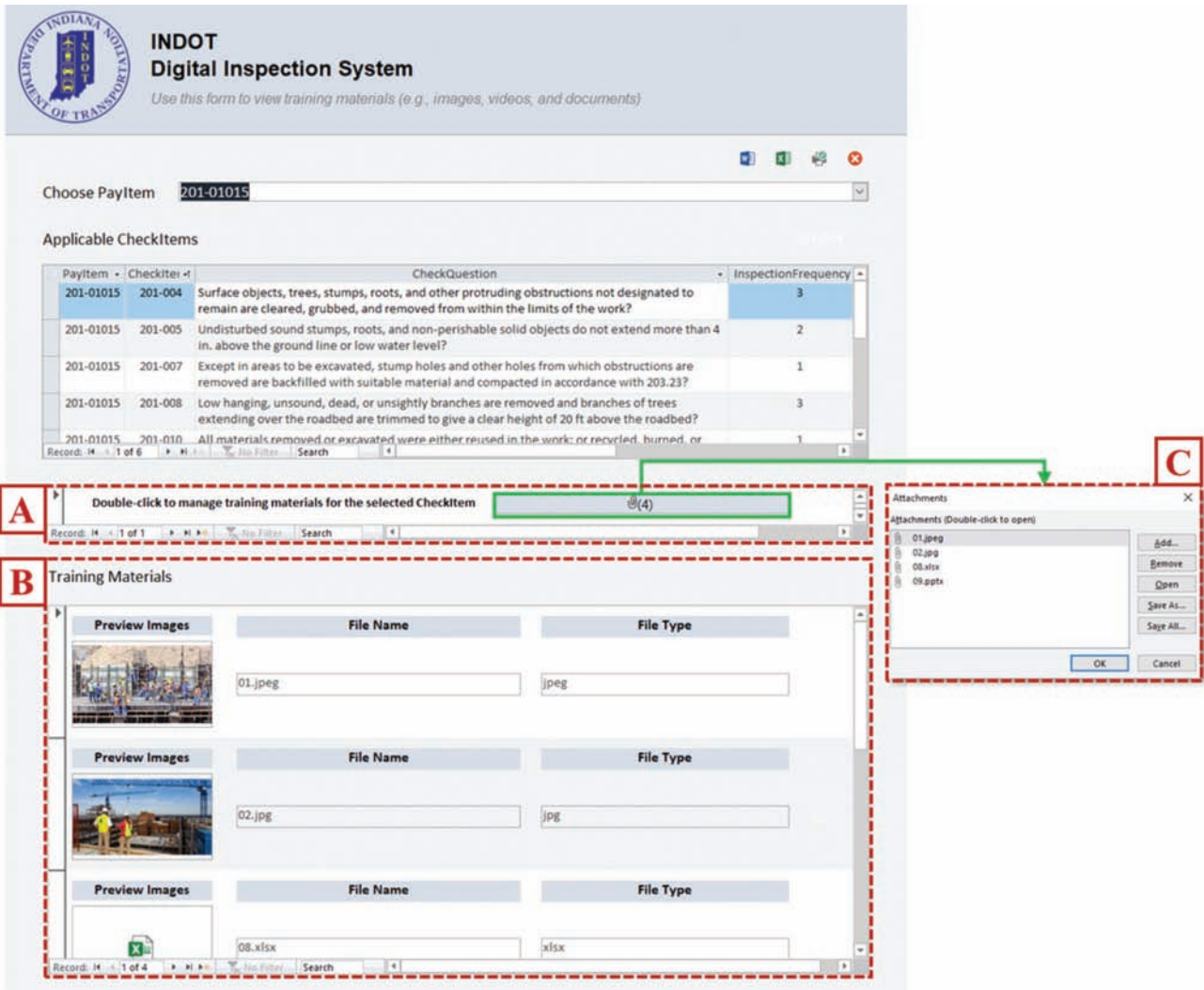
Figure 6.3   Sample illustration: Embedding files in database.
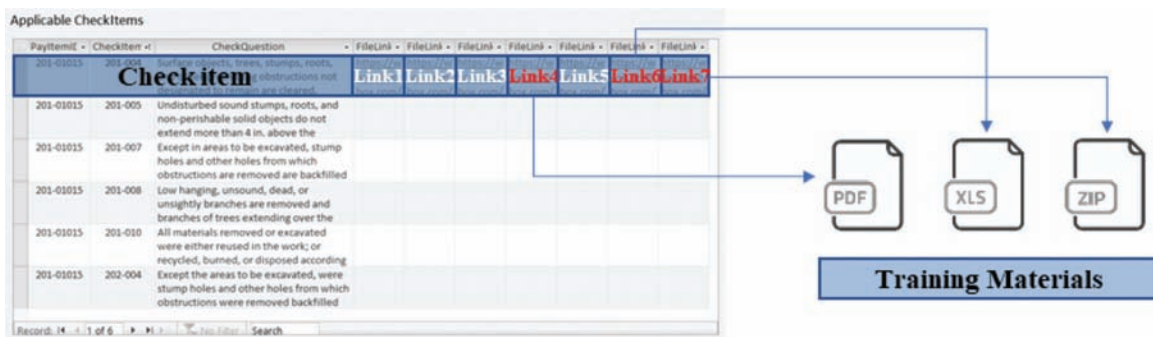
Figure 6.4   Mechanism 3.1: Link to individual files directly.

- For Mechanism 3.2, each check item is connected to one folder, which makes the data structure straightforward. However, a file may be relevant to multiple check items, which causes duplicate file issues. A variant of Mechanism 3.2 is linking every check item to a single "linking file" (in place of a folder) that contains varying numbers of links to files stored either locally or on Cloud.

6.5 Summary and Suggestions

Table 6.1 summarizes the advantages and disadvantages of the mechanisms. The recommendation for INDOT is to use Mechanism 1 for websites and the variant to Mechanism 3.2 for files.

Figure 6.5 Mechanism 3.2: Link to folder.

Figure 6.6 Sample illustration for mechanism 3.1.

Figure 6.7 Sample illustration for mechanism 3.2.

TABLE 6.1
Comparison of Mechanisms

	1. URL Link to External Websites	2. Embedding Files in the Database	3. Organizing Materials in Folders and Linking to:	
			3.1 Individual Files	3.2 Explorable Folders
Pros	Minimal local storage requirements	Full control of the materials Direct views of the materials without further retrieval	Easy and quick retrieval	Easy to manage and update
Cons	No control of the materials Wi-Fi connection issue	Large storage space requirements Less flexibility to update DB	Unable to see relative files Data structure issue	Needs further development to retrieve files

7. AUTOMATE THE GENERATION OF NEW CHECKLISTS AND COMPLIANCE CHECKING

7.1 Introduction

Specifications change from time to time and unique special provisions may be necessary for specific projects. The manual process of extracting check items from these textual documents is time-consuming, and the results may become outdated quickly. The objective of this task was to automate the generation of a new checklist using natural language processing (NLP) techniques when there are specification changes. Figure 7.1 illustrates the vision for this objective.

The specific NLP technique that applies to this task is sentence classification whereby a sentence is classified as a requirement or nonrequirement. Sentence classification is a well-established technique in the NLP domain for various purposes (e.g., spam filtering, email categorization, document classification, and sentiment analysis). Figure 7.2 illustrates its typical process. Text preprocessing is the very first step to process the unstructured textual data, which includes noise removal, word normalization, and word standardization. After preprocessing, the processed data are converted into features such as term frequency, tokenization, and vectorization as input to classification models. The classification models are trained and then used to predict whether or not a sentence is a requirement based on the input features.

7.2 Sentence Classification

The objective of sentence classification in this project is to identify the inspection requirements (e.g., check questions) from the specification, which generally consists of numerous sentences. Figure 7.3 illustrates the workflow for sentence classification: (1) individual sentences are extracted from the specification document; (2) the resulting sentences are preprocessed to remove spaces, stop words, and punctuations; (3) Word2Vec and GloVe algorithms are used to extract the vector features from the cleaned sentences; and (4) the machine learning algorithms of the convolutional neural network (CNN) and recurrent neural network (RNN) models are developed as a classifier. Experiments were conducted to test the different models and different training/validation/testing strategies.

7.2.1 Dataset Preparation

The PDF file of the *INDOT Standard Specification 2018* (INDOT, 2018) was used as the source file to develop the dataset for NLP exploration. Figure 7.4 (left) illustrates that the file structure is a hierarchy of subsections-section-division, and the requirements are expressed as sentences under the corresponding subsections. The table of contents (TOC), shown on the right side of Figure 7.4, clearly outlines the hierarchy. Regular expression (RE), which is a text passing technique, was applied to identify the subsection-section-division hierarchy in the file and to separate the textual contents correspondingly. Specifically, the TOC provided patterns and the RE technique identified and extracted the text following the patterns. For example, a section heading followed the format of "Section 000 - XXX" in the specification, and the sample RE code below extracted all the section headings.
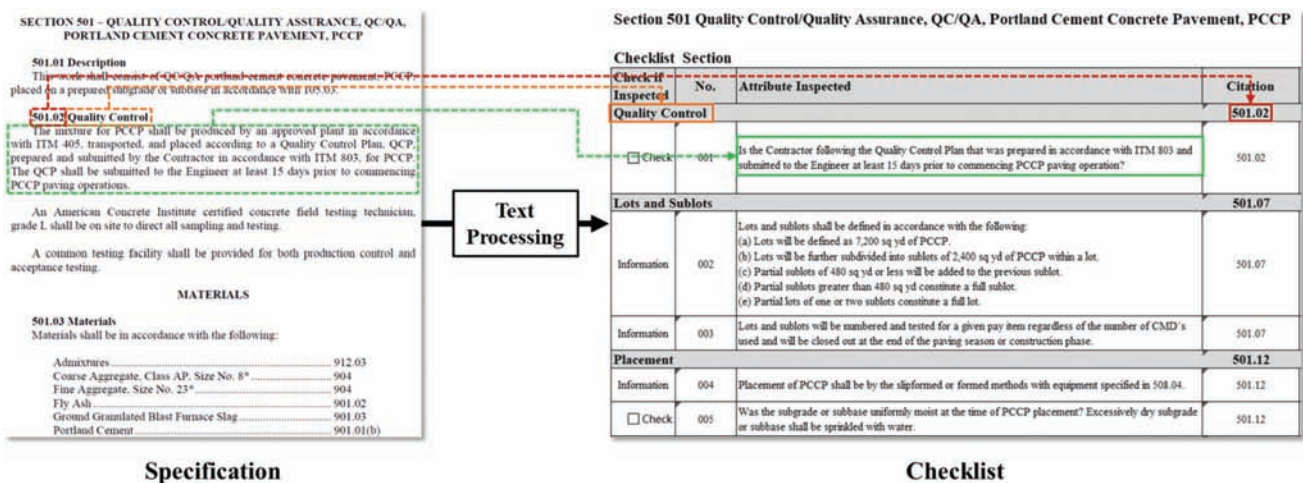
Figure 7.1 Automatic extraction of check items and matching with pay items.

Figure 7.2 Illustration of sentence classification process.
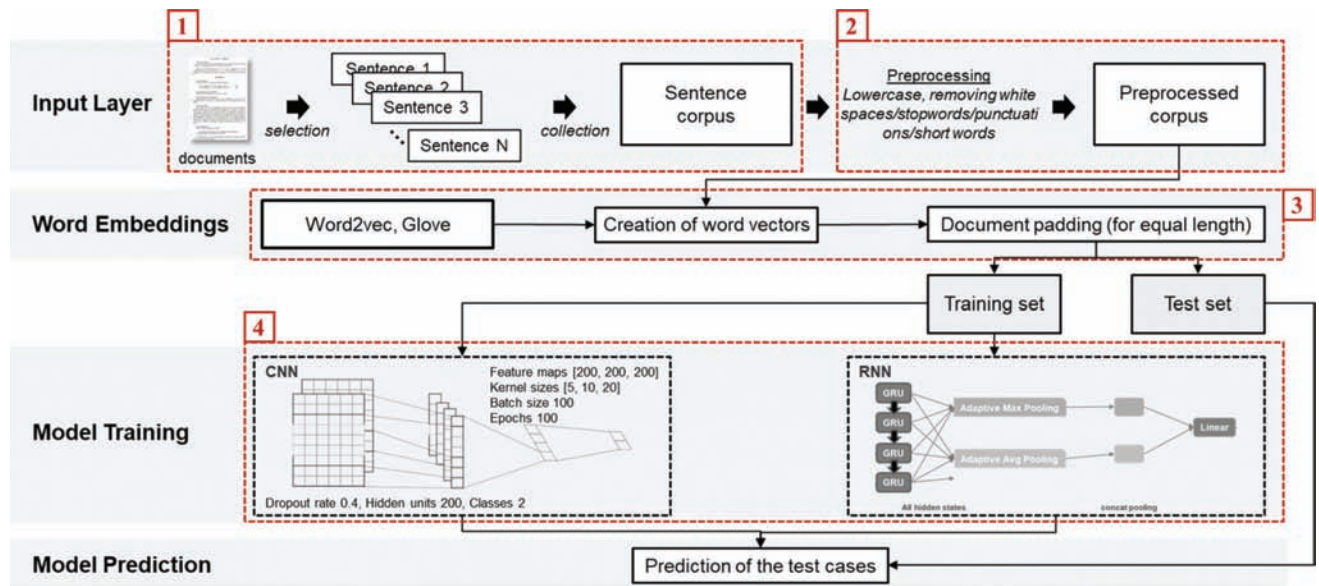
Figure 7.3 Workflow for sentence classification.
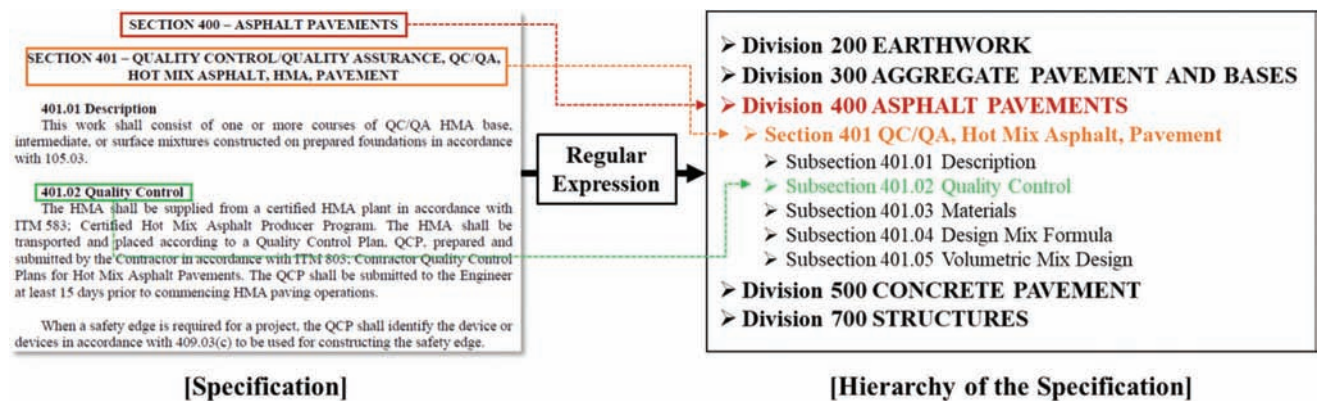
Figure 7.4 Development of specification hierarchy.

Pattern Outter_pattern = Pattern.compile("SEC-TION ([0-9]{3})(.*)\r\n"); Pattern Inner_Pattern = Pattern.compile("([0-9]{3})\\.([0-9]{2})(.*)\r\n(.*) [0-9]{2}");

A total of 3,171 sentences were extracted based on the ".", space, and return characters. These sentences were grouped under subsections. The sentences included in the inspection forms as check items of the type "Check" were labeled with "1." All the remaining sentences were labeled with "0," indicating they were nonrequirements. Figure 7.5 illustrates this labeling procedure with a requirement example. 1,921 sentences out of the 3,171 were determined to be requirements and were labeled with "1."

7.2.2 Data Preprocessing

Most of the sentences contain words (e.g., is, am, the, and, of); symbols (i.e., punctuation); and phrases (e.g., URLs) that were irrelevant for sentence classification.

They were removed in this step using several techniques (e.g., tokenization, removing stop words, capitalization, and stemming). Figure 7.6 illustrates the data preprocessing step with a sample sentence. After removing irrelevant words and symbols and converting all the uppercase letters to lowercase, a sentence was broken down into a set of words called tokens.

7.2.3 Feature Extraction and Classification

The preprocessed data in the format of tokens needed to be converted into features to be used as input in the sentence classifier (e.g., CNN and RNN models). Word embedding is a technique that redefines high-dimensional word features into low-dimensional feature vectors by conserving the contextual similarity in the corpus. Word2Vec and GloVe are two widely used word embedding algorithms. Both algorithms provide a vector for every word in the corpus based on their co-occurrence information (i.e., how frequently
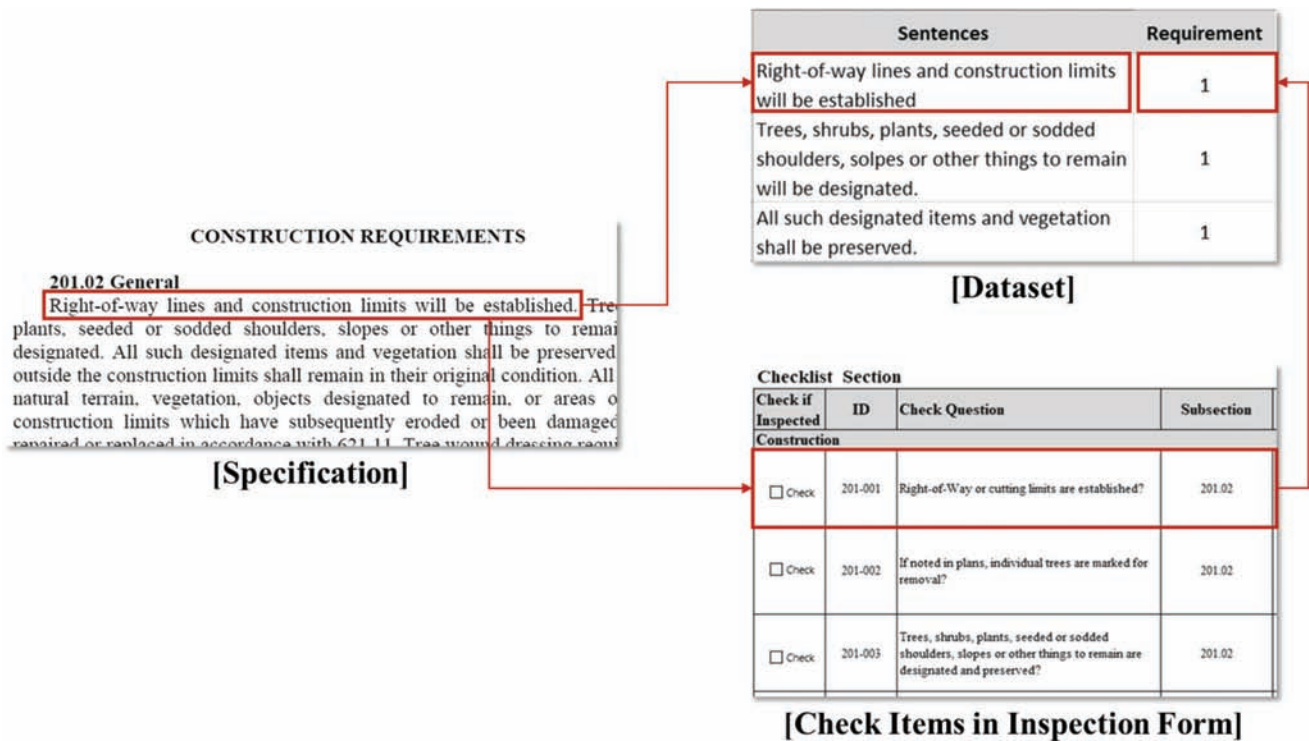
Figure 7.5 Dataset development.

Figure 7.6 Illustration of preprocessed data.

TABLE 7.1
Classification Results Based on Different Word Embedding Algorithms

Division	Word Embedding	Training Results		Testing Results	
		Accuracy[1]	Loss[2]	Accuracy	Loss
CNN	Word2Vec	0.7855	0.3865	0.8016	0.4059
	GloVe	0.8697	0.3089	**0.8800**	**0.3851**
RNN	Word2Vec	0.8475	0.1866	0.7123	0.6125
	GloVe	0.9323	0.1763	0.7900	0.5723

[1]Accuracy represents the number of correct predictions divided by total number of predictions.
[2]Loss represents a difference between the predicted value and the true value.

the words appear in the corpus). The main difference between them is that Word2Vec is a predictive model while the GloVe is a count-based model. In this project, both algorithms were tested for their ability to convert tokens into feature vectors. The resulting feature vectors served as input to two machine learning algorithms to train and test the classification model. The two specific machine learning algorithms tested and assessed in this study were CNN and RNN, based on their proven performance in sentence classification.

The 500 sentences extracted from the Division 500 sections in the *INDOT Standard Specification 2018* were used as the training data, following an 80/20 training/testing strategy. Four hundred sentences were randomly chosen for training the models, and the remaining 100 sentences were used to test the trained models and assess their performance. Table 7.1 presents the results for varying combinations of the classification models and word embedding algorithms. The CNN + GloVe combination produced the highest accuracy and

TABLE 7.2
Classification Results for Division 200, 300, 400, and 700

Division	Number of Sentences	Testing Results			
		Accuracy	Loss	Precision	Recall
200	400	0.82	0.36	0.88	0.88
300	132	0.74	0.63	0.85	0.77
400	540	0.87	0.28	0.90	0.91
700	1,600	0.78	0.31	0.82	0.89

TABLE 7.3
Classification Results of Additional Experiments

Trials	Dataset			Testing Results			
	Training (%)	Validation (%)	Test (%)	Accuracy[1]	Loss	Precision	Recall
Trial 1	50	20	30	0.86	0.27	0.93	0.89
Trial 2	50	0	50	0.84	0.33	0.89	0.88
Trial 3	40	0	60	0.81	0.31	0.85	0.87
Trial 4	30	0	70	0.68	0.49	0.77	0.74

[1]Accuracy represents the average accuracy for each section (using test dataset).

the lowest loss. Therefore, this combination was used in further experimentation and analysis.

7.2.4 Experimentation and Testing Results

A series of experiments were conducted to test the performance of the CNN + GloVe combination. Table 7.2 shows the performance of the CNN + GloVe model trained using the Division 500 sentences on Divisions 200, 300, 400, and 700. The accuracy measure was the number of correct predictions out of the total number of predictions. On average, the model achieved 80% accuracy. Loss is the value of the loss function that describes the fit of the trained model to the data. Loss ranges from 0 to 1; and a smaller loss value indicates a better fit of the model to the data. Precision and recall were the two metrics that were used to measure the effectiveness of the model. Equation 7.1 illustrates their calculation, where TP is the number of true positive (requirement sentences correctly classified), FP is the number of false positive (nonrequirement sentences classified as requirement), and FN is the number of false negative (requirement sentences incorrectly classified as nonrequirement). In this project, precision was the fraction of sentences classified as requirements that were truly requirements, and recall was the fraction of sentences classified as positive that were correctly classified.

$$Precision = \frac{TP}{TP+FP} \quad Recall = \frac{TP}{TP+FN} \quad \text{(Equation 7.1)}$$

Additional experiments were conducted to train the model using sentences extracted from each division. Table 7.3 summarizes the results of the four different trials. For example, in trial 2, 50% of the total sentences from each division were used to train the model, and the remaining 50% were used to test the trained model. In trials 1, 2, and 3, it was observed that training the model using sentences extracted from each division led to higher accuracy compared to the accuracy of 80% achieved by using Division 500 as the training data set.

7.3 Findings and Recommendations

The main findings were as follows:

- The CNN model and GloVe algorithm can be used as tools to develop a classifier to identify the inspection requirements from the specification.
- Using training samples from all the divisions, a higher percentage of data for training, and a validation data set led to better classification performance.
- Validation helped optimize the hyperparameters of the model (e.g., number of epochs, batch size, and hidden layers).
- The 3,171 sentences extracted from the *INDOT Standard Specification 2018* labeled with the correct classes (i.e., "1" for requirement and "0" for nonrequirement) provided a valuable labeled dataset to train models and automatically extract the inspection requirements from other versions of the specification.

8. SUMMARY AND RECOMMENDATIONS

8.1 Summary

In this project, a relational database and toolset were developed to automatically generate customized checklists of construction requirements at the pay item level to replace the manual, time-consuming, and error-prone process of gathering construction requirements

from textual documents in the current practice, which will enhance the efficiency of the construction inspection process at a time when INDOT is challenged by resource shortages. This project's achievements are summarized below.

- A maintainable database was designed and constructed to store check items and link them to applicable pay items using a linking mechanism. The corresponding user interface was implemented in MS Access, using forms, SQL statements, and VBA scripts. A business process also was designed to update the database when there are specification changes.
- Of all the 2,220 pay items, 1,116 were identified as having repetitive workflows; and a total of 29 construction processes with stages involved in the corresponding inspection process were identified. Database mechanisms were designed to associate the pay items to their corresponding workflows and the group check items to their corresponding construction stages through subsections.
- Among all the 2,415 check items, 304 were identified that cross-reference other subsections. A linking mechanism was developed to facilitate the navigation from referencing check item to check items under the referenced subsection.
- Further customization and subgrouping functionalities were achieved by using MS Access tools and VBA code.
- Four mechanisms to associate multiple documents in varying formats (e.g., photo, video, Word, Excel, PDF, websites) were explored, tested, and assessed. The advantages/disadvantages of each mechanism were identified and recommendations were made correspondingly.
- Natural language processing techniques (e.g., sentence classification, GloVe, CNN) were explored and tested in several experiments of varying scenarios.

The main deliverable is an intelligent database of checklists with user-friendly tools that turns out to be a powerful tool that generates customized checklists with sufficient details on demand to field inspectors, thereby reducing their workload and enhancing their inspection efficiency.

8.2 Implementation and Cost Savings

A total of three implementation strategies are feasible, namely, the direct use of the static inspection forms organized by specification sections, connecting the database to the field app., and the standalone use of the database (and its current interface). Following a meeting and several rounds of communications with the Management Information Systems Unit at INDOT, the recommendation is to (1) use the static forms immediately and (2) incorporate the database into INDOT's Enterprise database and develop connection to the field app. Through a separate INDOT project. Note the research team will only provide assistance as deemed necessary by INDOT.

The newly developed system and inspection forms are expected to assist field crew to gain both efficiency and efficacy, as proven in the pilot study. They also

eliminate all the guessing work and the need to manually search for guideline regarding what to check, how to check, what are the quality requirements, and how to deal with suspected noncompliance. Jointly, it reduces the risk of accepting low quality product, enhances the long-term performance of infrastructure, and lowers the maintenance and repair cost.

8.3 Findings and Recommendations

The main findings of this project were as follows:

- The database approach (including both the database design and the user interface) is capable of achieving the desired functionalities to complete the desired toolset.
- The division-section-subsection-requirements hierarchy will help organize the check requirements and the applicable pay items. Specifically, subsections were found to play a central role in linking the pay items, construction processes and stages, and check items.
- Testing the four mechanisms that could accommodate the inspection documents and training materials in varying formats revealed the following.

  ○ A URL link is sufficient to connect checklist items with websites.
  ○ Embedding files in the database is impracticable due to size constraints and data duplication.
  ○ Organizing files under folders and linking pay items to folders are efficient approaches, but they require additional development efforts to explore and navigate through the file system.
  ○ The variant version, wherein each check item is linked to a central linking file that contains links to all the applicable documents stored on the cloud, saves storage space and eliminates data redundancy.
  ○ Finally, each mechanism has advantages and disadvantages and a common challenge is to build and maintain all the links.

- NLP techniques can assist in automatic extraction of check items from the *INDOT Standard Specification 2018* with approximately 90% accuracy.

  ○ A regular expression (RE) was efficient in identifying the hierarchy (subsections-section-division) of a file and separating sentences based on textual patterns.
  ○ The sentence classifier, which consists of GloVe (a word embedding algorithm) and convolutional neural network (CNN) (a machine learning algorithm) showed the highest accuracy and the lowest loss.
  ○ Using training samples from all divisions, a higher percentage of data for training, and having a validation set led to better classification performance.

- The 3,171 sentences extracted from *INDOT's Standard Specification 2018* and labeled with correct classes ("1" for requirement and "0" for nonrequirement) comprised a valuable labeled dataset to train models and automatically extract inspection requirements from other versions of specification.

The research team for this project offers the following implementation recommendations:

- The database designed in this project is recommended to INDOT for managing their inspection system and generating customized inspection checklists.
- To accommodate photos/videos of inspection documentation and training materials, it is recommended that INDOT implement Mechanism 1 (using URL link) for external websites and a variant version of Mechanism 3.2 for files in varying formats assumed to be stored on cloud drives.
- For automatic extraction of check items from the specification, especially different versions of the standard specification, it is recommended that INDOT adopt the NLP-Machine Learning approach (CNN+GloVe).
- Testing the inspection database in a field application is recommended prior to full-scale implementation.

REFERENCES

INDOT. (2018). *Indiana Department of Transportation 2018 standard specifications*. Indiana Department of Transportation. https://www.in.gov/dot/div/contracts/standards/book/sep17/2018Master.pdf

Xu, X., Yuan, C., Zhang, Y., Cai, H., Abraham, D. M., & Bowman, M. D. (2019a). Ontology-based knowledge management system for digital highway construction inspection. *Transportation Research Record*, *2673*(1), 52–65. https://doi.org/10.1177/0361198118823499

Xu, X., Zhang, Y., Yuan, C., Cai, H., Abraham, D. M., & Bowman, M. D. (2019b). *Risk-based construction inspection* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2019/06). https://doi.org/10.5703/1288284316916

About the Joint Transportation Research Program (JTRP)

On March 11, 1937, the Indiana Legislature passed an act which authorized the Indiana State Highway Commission to cooperate with and assist Purdue University in developing the best methods of improving and maintaining the highways of the state and the respective counties thereof. That collaborative effort was called the Joint Highway Research Project (JHRP). In 1997 the collaborative venture was renamed as the Joint Transportation Research Program (JTRP) to reflect the state and national efforts to integrate the management and operation of various transportation modes.

The first studies of JHRP were concerned with Test Road No. 1—evaluation of the weathering characteristics of stabilized materials. After World War II, the JHRP program grew substantially and was regularly producing technical reports. Over 1,600 technical reports are now available, published as part of the JHRP and subsequently JTRP collaborative venture between Purdue University and what is now the Indiana Department of Transportation.

Free online access to all reports is provided through a unique collaboration between JTRP and Purdue Libraries. These are available at http://docs.lib.purdue.edu/jtrp.

Further information about JTRP and its current research program is available at http://www.purdue.edu/jtrp.

About This Report

An open access version of this publication is available online. See the URL in the citation below.

Cai, H., Jeon, J., Xu, X., Zhang, Y., & Yang, L. (2020). *Automating the generation of construction checklists* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2020/23). West Lafayette, IN: Purdue University. https://doi.org/10.5703/1288284317273