



W912HZ-14-C-0026

Information Dredging

Task 1: Historic Information Dredging

Chief's Report Processing Design and Results

Task 2

Deliverable: Option 2 Final Report

Document Number: W912HZ-14-C-0026-Extraction-201709

Document Date 29 September 2017

Historic Information Dredging

Chief's Report Processing Design and Results

Prepared for Army Corp of Engineers by:

Vencore Labs

Technical contact:

John R. Wullert II
908-748-2687
jwullert@vencorelabs.com

Trademark Acknowledgments

All brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Change History

Issue	Date	Revised By	Description of Change	Page(s)
V1.0	9/29/16	John Wullert	Initial Release of the document	All
V2.0	9/29/17	John Wullert	Update to reflect enhancements and improvements associated with the second version of the process and delivered data.	Changes throughout. Specific addition of Section 2, Section 4.1.3. and Section 6.4

Stakeholders

Name
Tanya Beck
Gregory Dreaper
Kenneth Ned Mitchell
Julie Rosati
Brandan Scully

Table of Contents

1. Introduction	1
2. Keys to Data Interpretation	2
2.1 Data Quality Detection	2
2.2 Repeated Data Identification	4
3. Data Model.....	6
3.1.1 Data Modelling Concept.....	6
3.1.2 Data Model Development.....	6
3.2 Data Model	7
3.2.1 ACE_District Entity	8
3.2.2 ACE_Project Entity	8
3.2.3 ACE_NavigationPath Entity.....	9
3.2.4 ACE_Operation Entity	10
3.2.5 ACE_Placement Entity.....	12
3.2.6 ACE_Removal Entity	13
3.2.7 ACE_Date.....	14
3.2.8 ACE_Document	15
4. Data Validation and Testing	16
4.1 Data Validation Rules and Results	16
4.1.1 Validation Rules	16
4.1.2 Validation results (Version 1.0).....	17
4.1.3 Validation results (Version 2.0).....	21
5. Data Delivery Format	25
5.1 Data File Format.....	25
5.1.1 Operation File.....	26
5.1.2 Placement File	27
5.1.3 Removal File	28
5.2 Proposed Database Schema	29
6. Information Extraction Design Details	35
6.1 Document Format and Information	35
6.1.1 Document Date	35
6.1.2 Document Volume and Part Numbers.....	36
6.1.3 Line Feeds and Multi-Spaces	36
6.1.4 Numbers	37
6.1.5 Extraneous Split Removal	37
6.1.6 Hyphen Removal	38
6.1.7 Other Dividers	38

6.1.8	Page Headers and Page Numbers	39
6.2	Information Elements	41
6.2.1	Project.....	41
6.2.2	Navigation Path	44
6.2.3	Placement Structures	47
6.2.4	Volume or Weight Removed and/or Placed	47
6.2.5	Operations Characteristics	50
6.2.6	Multi-part Operations	55
6.2.7	Districts.....	56
6.3	Storing Results.....	58
6.4	Galveston Processing.....	59

1. Introduction

The following is an example paragraph taken from 1965 Annual Report of The Chief of Engineers on Civil Works Activities:

Operations and results during fiscal year. Under a contract for dredging channel north of Shooters Island completed last fiscal year, an adjustment was made decreasing costs by \$3,719 for maintenance. U.S. seagoing hopper dredge Essayons and attendant plant were employed intermittently from July 16 to August 23, 1964, in dredging to restore project depths in Raritan Bay Channel. Removed 367,500 cubic yards, place measurement, of material at a cost of \$341,489, including \$4,241 for engineering preliminary to dredging for maintenance.

This paragraph contains many information elements that describe this particular set of operations. These information elements include:

- the fact that the operations occurred during the current fiscal year,
- the amounts and types of material that were removed (367,500 cubic yards of material),
- the means by which the quantity was determined (place measurement),
- the navigation path that was the subject of the work (Raritan Bay Channel),
- the start and end dates (July 16 to August 23, 1964) and
- the cost of the operation (\$341,489).

The information from this paragraph does not represent the full picture, however. For example, there is no mention of the project/location of the operations nor the District responsible for this project. This information (New York and New Jersey Channels, in the New York, NY District) is found elsewhere in the document.

The aim of the Information Dredging project was to extract this type of key information regarding dredging operations from instances of the Chief of Engineers' reports. The information extraction was accomplished through customization of a natural language processing tool. This customization was guided by a data model that specifies the information elements that are of interest. These information elements included specific entities, the relationships between those entities and the attributes that define them.

This document is organized as follows. Section 2, newly added in version 2.0 of this document, provides key recommendations for interpreting and processing the extracted information. Section 3 provides a detailed description of the data model, including each of its parts and the rationale behind them, in order to provide a complete view of the project and the data that it produced. Section 4 describes the validation and verification process that was applied to the data and provides the results of that effort. Section 5 describes the format of the delivered data. And finally, Section 6 describes the structure and logic of the information process used to extract the information.

2. Keys to Data Interpretation

This section is added to the front of Version 2.0 of this document to make it easy to find. The intention is to highlight key features of the data that will help ACE SMEs obtain maximum value from the information. Note that for those with little prior exposure to the data, it may be necessary to read later descriptions of the data model to fully comprehend the material here.

2.1 Data Quality Detection

A key problem with any automated data extraction process is that of errors. Such errors come in two flavors:

- Precision errors, where the identified information is incorrect
- Recall errors, where desired information present in the document is not detected

These two types of errors are often traded off against one another: making the process more open and flexible to increase recall will result in the capture of more information incorrectly. The extraction process for the Chief's reports is not immune to this effect, but steps have been taken to limit its impact.

In particular, a variety of information elements have been provided to cross-check the accuracy of the identified values. The availability of these cross-checks made it possible devise more flexible rules to increase the recall of the recognition process, knowing that some of the corresponding precision errors could later be identified and handled. The following means of cross-checking are provided in the data.

Stated District. The StatedDistrict parameter captures a top level of the hierarchy for dredging operations. This information is captured from headings in the document that can be many pages away from the statements describing the operations. As such, the presence of this information is a good indication that the process has understood the structure of the document. Three aspects of the district information are provided to help ensure accuracy:

- **Missing District:** If the StatedDistrict parameter is blank, it is most likely an indication that the operation was described in the early portion of the document, ahead of any of the headers that specify the districts. These values tend to be summary values found in the overview section of the document and thus are not values associated with a particular project or navigation path. As such, records with a blank StatedDistrict parameter should be ignored. There are fewer than 100 such records in the dataset.
- **MRC/CDC Districts:** Many of the documents contain sections reported by the *Mississippi River Commission* and/or the *California Debris Commission*. These sections of the document used a different format for reporting their operations than the format used the main sections document. In some cases, it was observed that reporting in these sections duplicated reporting in other Districts. Based on prior discussions with Army Corp of Engineer subject matter experts, these sections were

deemed to be of lesser interest and the information extraction process was not optimized or tested extensively on content from these sections. In some cases, it was noted that operations described in these sections duplicated content in other sections of the document. As such, operations that have a StatedDistrict parameter of MISSISSIPPI RIVER COMMISSION or CALIFORNIA DEBRIS COMMISSION should not be accepted by default but should be assessed carefully before inclusion in any subsequent processing.

- **ProjectDistrict versus StatedDistrict:** The information extraction process tags identified and recognized projects with information about the district to which such projects are currently assigned. Cases where the ProjectDistrict parameter and the StatedDistrict parameter do not match therefore represent potential errors. This was a useful debugging tool for identifying cases where district headers were being missed and other similar issues. Issues detected in this manner were addressed by modifying the rules or correcting critical OCR errors in the data. The majority of remaining cases where the two do not match, which total fewer than 1200 of the nearly 33K Removal operations in the extracted data, can be attributed to one of two causes:
 - **Changes in Districts:** The movement of district boundaries over the years and the addition and deletion of districts have created cases where the current assignment of projects to districts does not match the assignment in earlier years. Examples of this include projects now in the Alaska District that were formerly in the Seattle District and projects that are now in the Philadelphia District that were formerly in the Wilmington District.
 - **Matching Project Names:** In some cases, there are projects of the same name in multiple districts. For example, there is an *Illinois Waterway* project in both the Rock Island District and the St. Louis District.

Given that these reasons explain the majority of mismatch cases, and these reasons do not represent errors with the data, these data should be accepted as valid, using the StatedDistrict and ignoring the ProjectDistrict.

Project: Like the Stated District, the ProjectName parameter captures a portion of the Army Corp of Engineers organizational hierarchy, and the presence of this information is another indication that the information extraction process has understood the structure of the document. There are approximately 140 Removal operations that lack a ProjectName. Many of these have empty StatedDistrict parameters as well. Records with a blank ProjectName field should be ignored.

Work Period: The WorkPeriod parameter is designed to capture when the operation took place, for example, whether it was during the current fiscal year, a condition at the end of the fiscal year, part of an existing project, proposed future operations or work performed by local cooperation. The WorkPeriod parameter is populated for 82% of the removal operations. The fact that the value is not populated is an indication that the extraction process identified the operation in an unusual situation and thus it is more likely to be an error. The majority of cases where the WorkPeriod parameter is not populated occur in two types of situation:

- Documents from the early years (pre-1915) were less explicit about indicating when work occurred. This accounts for 6.8% of the Removal records with the WorkPeriod

attribute not populated. As described below, the start/end and in-progress dates provide an alternate means of determining which of these activities occurred during the current fiscal year.

- Mississippi River Commission and California Debris Commission sections of the document. These sections were written using a different format that is not handled as well by the extraction process, as described above. This accounts for 6.2% of the records with the WorkPeriod attribute not populated.

After these two factors are accounted for, only 4.8% of the Removal records have an unexplained unpopulated WorkPeriod attribute.

Records marked as “During Fiscal Year” are the primary records of interest and can be separated from those in other categories. In addition, a secondary check is available. Many records have an associated start/end date or in-progress date that capture specific date ranges for the operations. This date information is not found in the Removal records, but in the associated Operation records. (Note: Each Removal record has a pointer (foreign key) to the associated Operation record.) Removal records that lack a WorkPeriod parameter but that have corresponding start/end or in-progress date information that fits into the current fiscal period should be treated as occurring during the fiscal year. Records with neither a WorkPeriod as “Current Fiscal Year” or dates that fit within the current fiscal year should be not be treated as occurring during the reporting time period.

2.2 Repeated Data Identification

As was discovered during the assessment of the first delivery of the data set and in subsequent analysis, there are a few ways in which information can be repeated within the documents, raising the risk of double counting.

Appendix: Material in the appendices of documents was found in several cases to repeat values in the main sections. It was not clear that this was universally true, but it was common enough to warrant tagging. Records with the Appendix attribute set to “yes” are those that were extracted from appendices of the documents. There are approximately 730 such Removal records. These records should be treated with some care, comparing their values to others from the same District/Project to determine if they are duplicates. (Note: in later years, the appendices were in separate documents and the process was not applied to these documents.)

Constituents: In certain cases, the text explicitly lists values and the sum of those values. For example:

Between October 4 and November 20, 1946, and from March 17 to 21, 1947, the U. S. hopper dredge Hoffman removed 219,230 cubic yards from shoals in the outer end of the 300-foot channel, and 42,000 cubic yards from the ocean-bar channel, a total of 261,230 cubic yards

The processing was enhanced to recognize this and other similar situations and to separately mark the sum (TotalValue = yes) and the Constituent parts (Constituent =

yes). The processing also includes a pointer (foreign key) from the Constituent record to the corresponding TotalValue record. To avoid double counting, one should remove the values where TotalValue = yes or remove the values where Constituent = yes. There are just over 400 Removal records with either Constituent or TotalValue equal to yes.

Note that in some cases, the sum of the constituents does not equal the total value. For example, in the following text:

The U. S. dredge Henry Bacon, between March 20 and 28, 1941, removed 4,587 cubic yards of material, including 1,500 cubic yards of rock.

The 1,500 cubic yards is the only constituent of the 4,587 cubic yards. In cases where the constituents have a sum that is significantly different than the TotalValue, the TotalValue is more likely to represent the results of the actual operation. In many such cases, there is only a single constituent and thus they can be identified and handled appropriately.

The Constituent/TotalValue processing identified over 288 million cubic yards in duplicate Removal operations within the Current Fiscal year and over 1.7 billion cubic yards of duplication within the end of Fiscal Year reporting.

Repetitive Reporting: It was noted during the validation process that the values reported by certain districts were repeated exactly over a period of several years. For example, in the sections describing the project MISSOURI RIVER, KANSAS CITY TO THE MOUTH, the following identical text was used between 1941 and 1965, with occasional changes to the numeric values:

Condition at end of fiscal year.-The existing project was about 95 percent completed at the end of the fiscal year. ... dredge fills totaling 1,453,332 cubic yards, removal of 432,664 cubic yards of rock, removal of 81,081,860 cubic yards of material by dredging,

As an example of the types of changes that were observed, the value of 81,081,860 was used in 1941-1945, increased to 81,382,936 for 1946-1953 and increased again to 82,805,096 for 1955-57 and 1960-65. Thus the volumes being reported in this section are cumulative rather than per-year values.

The Removal operations within the example paragraph would be tagged with a WorkPeriod of "End of Fiscal Year", so these operations would not be counted as part of the current fiscal year activities. Thus the WorkPeriod attribute provides some protection against double counting of this repeated/cumulative reporting. Given the potential for errors in the assignment of WorkPeriod, as described above, it is recommended that the data be analyzed to detect volumes with values that are repeated across years, as well as for values that increase in a monotonic fashion over a period of years. This will provide a secondary check for such repetition and accumulation in reporting.

3. Data Model

3.1.1 Data Modelling Concept

In order to effectively capture information of the type described above, it is necessary to define a framework that represents the information. Such a framework, or data model, defines the primary information elements, the key attributes that describe or define those elements and the relationships between them. For example, a key conclusion that can be drawn from the paragraph above is that there was one specific activity – the dredging to restore project depths. This removal operation is a primary information element and is defined as an *entity* in the model. Associated with the removal operation entity are critical pieces of information that serve to define and specify the operation. These data elements include the type and amount of material that was removed, the units used to quantify those amounts and the means used to measure those quantities, along with the navigation path from which the material was removed. These *attributes* will be included in our model to describe, define and differentiate the entities. Finally, the removal operation took place under a specific project and the project is managed within a particular geographic district. To capture this hierarchy, Project and District will be separate entities in our model, with each project being associated with a district and each removal operation being associated with a project. In this way, the model provides a means of specifying the information that needs to be extracted and of applying structure to the results of the extraction process.

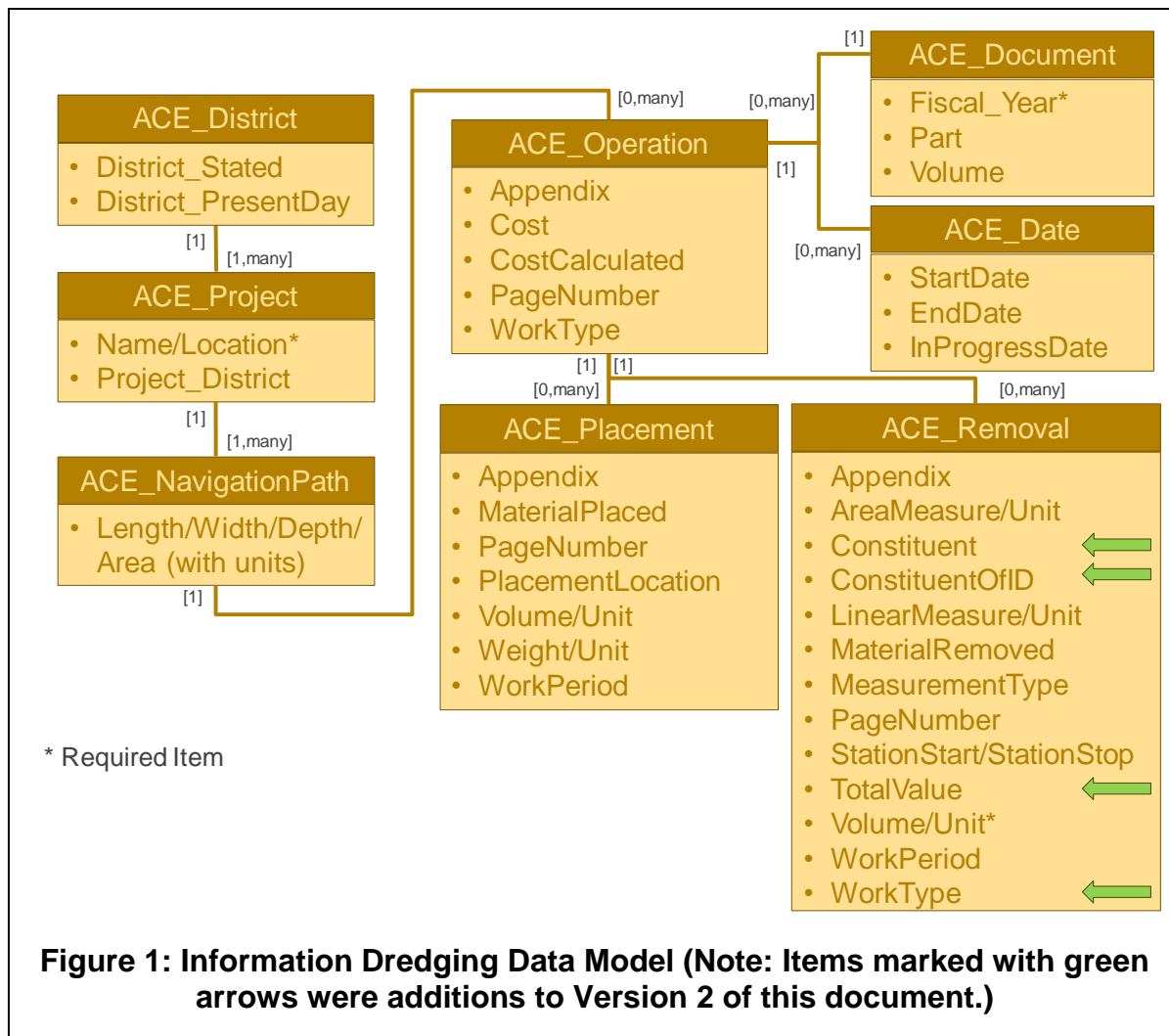
The following section describes the complete data model that was defined for this project and describes each of the entities, attributes, and relations.

3.1.2 Data Model Development

The data model used in this project was developed in an iterative fashion as a collaboration between the data scientist at ACS and the subject matter experts from the Army Corp of Engineers. The process started with the subject matter experts manually annotating sections of Chief's reports, highlighting the information that was of interest. Based on that input, ACS proposed an initial model. After discussion and refinement, ACS implemented rules to capture the data represented in the model. In subsequent interactive sessions and email exchanges, the team would review sample data that was extracted from the reports, discuss any anomalies or unexpected information uncovered during the processing and determine what data needed to be incorporated into the model. ACS would then refine the model and the processing rules to align with the updated model. This collective effort, driven by the data and characterized by regular feedback, was used in order to ensure that the model represented the most important information and that its structure would make it useful for subsequent analysis.

3.2 Data Model

The Information Dredging data model is shown in Figure 1. The model consists of eight entities, represented by the separate rectangles with their names in the light-on-dark text at the top. The lower portion of each entity rectangle, using dark-on-light text, lists the attributes associated with each entity. The relationships between the entities are represented by the lines between them and the numbers in brackets at each end of the lines. In this case, the relationships are not named, but the numbers specify cardinality constraints on those relationships, indicating the minimum and the maximum number of each entity that must/can be involved in the relationship.



The following subsections will focus on each of the entities, describing the concept, the attributes and its relation to other entities.

3.2.1 ACE_District Entity

For purposes of management of the operations that are of interest in this project, the Army Corp of Engineers has defined a hierarchical structure based on geography. This structure starts with geographic Divisions, encompassing large regions of the continental United States. Most Divisions are further divided into geography-based Districts. The ACE_District entity represents this level in the hierarchy.

Much of this geography-based hierarchy is reflected in the structure Chief's Reports, which have major sections assigned to each District. Note that the documents are somewhat inconsistent in their representation of this data. For example, while most of the sections are devoted to Districts (e.g., New York, N.Y. District, Philadelphia, PA. District), some are reported at the Division level, notably the New England Division. Given that the model was defined to represent the data extracted from the documents, this distinction between District and Division is not represented. Specifically, the New England Division is captured as an ACE_District entity.

The ACE_District entity has two attributes:

- District_Stated: name of the District as specified in the document
 - The text used to populate this attribute is normalized to common text for each of the known districts. For example, any varied references, such as "NEW YORK, N.Y. DISTRICT," or "NEW YORK DISTRICT", are all mapped to "New York District". The mapping operation that produces this normalization provides robustness against variations in the text format and to optical character recognition errors and the resulting normalized text should enhance reconciliation of district information across documents.
- District_PresentDay: district currently associated with that geographic region and associated projects
 - The set of districts has changed over time, which some districts being merged into others. For example, the "Washington, DC District" is no longer used and has been incorporated into the Baltimore District. Where such mappings from old to new districts could be identified, this information was included in reference data and used to populate this attribute.

The ACE_District entity has one relationship.

- ACE_Project: Each ACE District entity can be related to multiple ACE_Project entities, representing the many active projects that occur within each district in a given year. Note that an ACE_District entity will not be instantiated unless there is at least one ACE_Project of interest detected.

3.2.2 ACE_Project Entity

Within each district, work is managed via a set of Projects. There are multiple types of projects, including navigation projects, erosion control projects, and flood control projects.

Navigation projects, which are the projects of primary interest in this task, are generally associated with a specific body of water, a segment of a body of water or a set of such bodies. For example, within the Philadelphia District, the set of Navigation Projects includes

- Absecon Inlet, N.J.,
- Delaware River, Pa., N.J. and Del., Philadelphia to the sea
- Indian River Inlet and S Bay, Del.

The ACE_Project entity represents this level of the management hierarchy.

The ACE_Project entity has two attributes:

- Name/Location: name of the project
 - This is the exact text listed in the document that specifies the project. This attribute is required and will be present in all cases
- Project_District: name of district that should be associated with this project
 - Reference data made available by the USACE SMEs included a list of current projects. This reference data included the district associated with each project. In cases where the project cited in a document can be linked to a project included in the reference data, the corresponding District information from the reference data is added as an attribute to the ACE_Project entity. This provides a means of cross-checking the accuracy of the associated District entity. Note that the assignment of projects to districts has changed over time, so a mismatch between the Project_District attribute and the District_Stated attribute of the associated ACE_District entity does not always indicate an error in the data.

The ACE_Project entity has two relationships.

- ACE_District: Each ACE_Project is associated with one and only one ACE_District.
- ACE_NavigationPath: Each ACE_Project entity can be related to multiple ACE_NavigationPath entities. Note that an ACE_Project entity will not be instantiated unless there is at least one ACE_NavigationPath of interest detected.

3.2.3 ACE_NavigationPath Entity

Each project can encompass one or more specific navigation paths. Such paths can include many types of waterways, including harbors, anchorages, rivers, and channels. The ACE_NavigationPath entity is used to represent these pathways.

This entity actually represents two aspects of these paths. In the first place, the ACE_NavigationPath entity is used to capture the characteristics of the navigation path, such as the length, width, and depth of a channel. In the second place, the ACE_NavigationPath entity is used to represent the subject of dredging operations. Ideally, these two uses would be met by the same entity instance. However, the characteristics of the path and the dredging operations are described in different paragraphs within the document with no clear and

consistent means of referral between them. For example, the Existing Project paragraph, which describes the characteristics, might refer to a

“channel 8 feet deep, 100 feet wide, and 950 feet long, through the bar at the entrance, and within the creek a channel 7 feet deep, 100 feet wide, and 2,500 feet long, with turning and anchorage basin 500 feet wide at upper end in vicinity of Colonial Beach waterworks”

while the Operations and Results During Fiscal Year paragraph might describe

“Maintenance dredging, by contract, to restore upper channel and turning basin to authorized dimensions.”

Reconciliation of the paths across these types of references is not possible without additional information. In addition, the natural language processing tools used for the extraction operation are not optimized for such operations.

The ACE_NavigationPath entity has eight attributes:¹

- Area: the reported area of the navigation path
- AreaUnit: unit of measure used for reporting the area of the navigation path
- Depth: the reported depth of the navigation path
- DepthUnit: unit of measure used for reporting the depth of the navigation path
- Length: the reported length of the navigation path
- LengthUnit: unit of measure used for reporting the length of the navigation path
- Width: the reported width of the navigation path
- WidthUnit: unit of measure used for reporting the width of the navigation path

Note that none of these attributes is required. If any of the characteristics is reported, the corresponding unit will be reported as well.

The ACE_NavigationPath entity has two relationships.

- ACE_Project: Each ACE_NavigationPath associated with one and only one ACE_Project.
- ACE_Operation: Each ACE_NavigationPath entity can be related to multiple ACE_Operation entities.

3.2.4 ACE_Operation Entity

During a given fiscal year, there may be zero or more tasks undertaken on each navigation path. These tasks can include individual placement or removal operations or combinations

¹ Note: The current rules, and the resulting data that has been delivered, do not attempt to capture the characteristics of the navigation path. Initial rules have been developed, but these have not been tested and refined to ensure quality of extract.

thereof. There are some information elements that are best associated with the entire task, regardless of how many specific activities make up the task. For example, the total cost of the operation, the type of work (e.g., new work, maintenance), and the dates of the work are all best associated with the task. The ACE_Operation entity is defined to serve as a means of linking these data elements together. In addition, the ACE_Operation entity serves as an anchor point that relates placement and removal activities that are parts of the same operation.

The ACE_Operation entity has five attributes.

- Appendix: indicates information detected in a document appendix
 - A small number of the documents in the corpus have appendices in the same file as the primary content. During testing, a noticeable number of false matches of dredging operations were observed in appendices. This effect was observed too late in the process to fully address the issue, so the suspect matches were marked by setting the Appendix attribute to “yes”.
- Cost: the total dollar amount spent on the operation
- CostCalculated: indication that reported Cost attribute is a sum of values found in the document
 - In most cases, cost information described in the documents is reported as the total amount associated with an operation, which may include multiple removal and placement activities. This is why the cost was modeled as an attribute of the ACE_Operation entity. In some cases, however, individual costs per activity are reported. For example:
Maintenance dredging was performed by the U. S. hopper dredge Hains and U. S. bucket dredge Tompkins removing 54,462 cubic yards bin measure and 17,310 cubic yards scow measure, at a cost of \$35,089 and \$39,078 respectively.
These two removal tasks are part of a single operation, but individual costs are reported. In such cases, the Cost attribute is set to the sum of the values (35,089 + 39,078 = 74,167) and the Cost_Calculated attribute is set to “yes” to indicate the summing operation.
- PageNumber: page in the document on which the information is found
 - This attribute must be treated as a string to allow for the Section-Page format used in some documents.
- WorkType: type of work for the operation
 - There are four possibilities here: New Work, Maintenance, Environmental, and Mixed. When the “Mixed” is used, additional information on the WorkType should be present on the corresponding Removal records.

The ACE_Operation entity has five relationships.

- ACE_NavigationPath: Each ACE_Operation entity is associated with one and only one ACE_NavigationPath entity.
- ACE_Date: Each ACE_Operation can be associated with zero or more ACE_Date entities. This allows for one or more start, end and in-progress dates for each operation
- ACE_Document: Each ACE_Operation entity is associated with one and only one ACE_Document entity.
- ACE_Placement: Each ACE_Operation can be associated with zero or more ACE_Placement entities.
- ACE_Removal: Each ACE_Operation can be associated with zero or more ACE_Removal entities.

Note: while neither the ACE_Placement nor the ACE_Removal entity is required, but any ACE_Operation entity will have at least one of the two.

3.2.5 ACE_Placement Entity

The ACE_Placement entity represents operations where material is intentionally moved to a particular location. Such placement might be the direct result of the material being removed from some navigation path or might be associated with construction.

The ACE_Placement entity has ten attributes.

- Appendix: indicates information detected in a document appendix
 - See note under ACE_Operation
- MaterialPlaced: the substance that was moved to a specific location
- PageNumber: page in the document on which the information is found
 - This attribute must be treated as a string to allow for the Section-Page format used in some documents.
- PlacementLocation: the site to which the material was moved
- Volume: measure of the quantity of material, characterized by occupied space
- VolumeUnit: standard for measuring the quantity of material, characterized by occupied space (most often, cubic yards)
- Weight: measure of the quantity of material, characterized by relative mass
- WeightUnit: standard for measuring the quantity of material, characterized by relative mass
- WorkPeriod: indication of when activity took place, relative to document fiscal year

- This value is determined based on the heading of the paragraph where the information is reported. Possible Values: During Fiscal Year, End of Fiscal Year, Existing Project, Proposed Operations, Local Cooperation

Note: “Local Cooperation” does not quite fit the definition of this attribute, but this information is also identified by the paragraph header and is mutually exclusive with the other choices, so the set was combined.

- WorkType: type of work for the operation
 - There are three possibilities here: New Work, Maintenance, and Environmental

Note: None of the attributes are required. However, the ACE_Placement entity will not be instantiated unless either information on Volume or Weight and the associated unit is available.

The ACE_Placement Operation has one relation:

- ACE_Operation: Each ACE_Placement entity is associated with one and only one ACE_Operation entity.

3.2.6 ACE_Removal Entity

The ACE_Removal entity represents operations where material is intentionally extracted from a particular location.

The ACE_Removal entity has 17 attributes.

- Appendix: indicates information detected in a document appendix
 - See note under ACE_Operation
- AreaMeasure: measure of the region of space affected by the removal operations
- AreaMeasureUnit: standard for measuring the region of space affected by the removal operations (e.g., square feet, square yards)
- (ADDED) Constituent: yes/no indicator that a value is a portion of another value in the document. In many cases, this captures cases where portions and totals are reported, such as amounts for maintenance and new work as well as the sum of the two. It is also used to indicate partial values, such as cases where the portion of the total amount removed that is one type of material is reported. (If text clearly distinguishes the amount during the fiscal year from total amount under a multi-year contract, the total amount is not recorded.)
- (ADDED) ContituentOfID: pointer to the record ID of total value that the current record is a portion of. The record pointed to by the ConstituentOfID will have a TotalValue parameter set to “yes”.
- LinearMeasure: measure of the region of space affected by the removal operations, characterized by length along the navigation path

- **LinearMeasure Unit:** standard for measuring the region of space affected by the removal operations, characterized by length along the navigation path (e.g., feet, miles)
- **MaterialRemoved:** substance that was extracted from the specific location
- **MeasurementType:** means used to determine the amount of material that was removed (e.g., bin measurement, measured in place)
- **PageNumber:** page in the document on which the information is found
 - This attribute must be treated as a string to allow for the Section-Page format used in some documents.
- **StationStart:** waypoint along navigation path at which removal operation began
- **StationStop:** waypoint along navigation path at which removal operation concluded
- **(ADDED) TotalValue:** yes/blank indicator that the value is a sum of other values captured in separate records. (See **Constituent** and **ConstituentOfID**).
- **Volume:** measure of the quantity of material, characterized by occupied space
- **VolumeUnit:** standard for measuring the quantity of material, characterized by occupied space (most often, cubic yards)
- **WorkPeriod:** indication of when activity took place, relative to document fiscal year
See note under **ACE_Placement** entity
- **(ADDED) WorkType:** type of work for the operation
 - There are three possibilities here: New Work, Maintenance, and Environmental
 - This is used only in cases where there is constituent/total relationship and the constituent values have individual work type values.

The **ACE_Removal Operation** has one relation:

- **ACE_Operation:** Each **ACE_Removal** entity is associated with one and only one **ACE_Operation** entity.

3.2.7 **ACE_Date**

There are multiple dates that can be associated with an operation. The text can describe the date that operations began or ended, or provide multiple such dates for operations that were performed in separate stages. For operations that were not concluded during the fiscal year, the text might report the fact that the activity was ongoing as of a particular date. The **ACE_Date** entity represents these various dates.

The **ACE_Date** entity has three attributes.

- **EndDate:** date on which operations were concluded

- InProgressDate: data on which operations were still ongoing, usually at the end of the reporting period
- StartDate: date on which operations began

None of the individual attributes is required, but at least one will be populated in any instance of the ACE_Date entity.

The dates are generally reported as they are written in the document. This means that the results can vary widely in format (e.g., July 13, 1965, July 1965). The processing does perform some extensions. For example, if the text reports that the operation was conducted “between July 13 and August 25, 1965”, the StartDate will be reported as “July 13, 1965” and the EndDate will be reported as “August 25, 1965”

The ACE_Date entity has one relationship.

- ACE_Operation: Each ACE_Date entity is associated with one and only one ACE_Operation entity.

3.2.8 ACE_Document

The information extraction process functions in a document-at-a-time fashion. Given that the content of a document is limited to a single fiscal year, this is an appropriate means of operation. The information about which fiscal year each operation was performed in is thus represented by the document itself. In some years, particularly during the 1930’s and 1940’s, the content for each fiscal year was spread across two documents. To allow users to make use of the page number information that is captured, it is necessary to capture not only the fiscal year but also the volume and/or part number of the document. The ACE_Document entity represents this content.

The ACE_Document entity has three attributes.

- Fiscal_Year: the four-digit year representing the reporting period of the document. This will be present in every ACE_Document entity
- Part: specific segment of multi-part documents
- Volume: specific segment of multi-volume documents

4. Data Validation and Testing

4.1 Data Validation Rules and Results

While the size of the corpus used in this project does not approach the realm of “big data”, the quantity of data extracted from and the page count of the approximately 100 files still create a dataset too large for complete manual verification. In order to ensure the quality of the result, it was necessary to define a set of validation rules that could be applied in a semi-automated fashion. The following subsections describe the validation rules and the results of the testing.

4.1.1 Validation Rules

The set of rules that were defined and applied fit into three categories, as described below.

The **Outlier Detection** validation rules were designed to verify the validity of outliers or entries with invalid forms or formats. This included rules to

- Identify volumes or weights that are extremely large
 - Removal volumes over 50,000,000 or placement volumes over 10,000,000
 - Placement weights over 10,000,000Verify that any values in these categories are valid.
- Identify volumes or weights that are extremely small
 - Removal or placement volumes less than 10
 - Placement weights less than 10Verify that any values in these categories are valid.
- Identify volume or weight values with invalid number formats
 - Removal volumes, placement volumes or placement weights that started with a “0”.Examine source document to determine cause of invalid number

The **Aggregation Comparison** validation rules were designed to verify temporal consistency in the number of operations that were detected and the amount of material involved.

- Sum amount of material (volume removed, volume placed) across all projects in each fiscal year. Ensure that the values in each decade do not differ by more than a factor of 10.
- Sum the number of removal operations and placement operations for each fiscal year. Ensure that the coefficient of variation (standard deviation normalized to average) is less than 25% for each decade.
- (ADDED) Fiscal year coverage: identification of temporal gaps in the reporting of removal operations. An automated assessment was used to identify fiscal years where there were not Removal operations reports. Manual investigation was then performed to evaluate cause/reasonableness of the gap.

The **Coverage and Consistency** validation rules were designed to ensure that the data was being obtained from most of the districts and that the stated district and project district matched.

4.1.2 Validation results (Version 1.0)

The validation rules described in the previous section were applied to the placement and removal records extracted from the entire corpus. The validation rules were used to identify possible problems, rather than to specifically identify the problem. Specific records or input documents that were identified by these rules were then investigated manually to determine if there was any errors or issues. When such errors were identified, suitable corrections were applied.

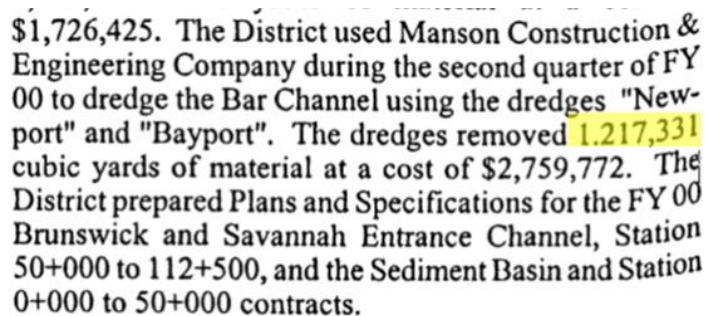
The analysis of large weights and volumes led to the discovery of a small number of optical character recognition errors, specifically where the space between two numeric values was omitted, causing them to be treated as a single number. An example of this type of error is:

Maintenance by contract: During the period Jul. 13 to Sep. 4,
1974,289,519 cubic yards of shoal were dredged from the lock forebay.

The lack of a space between 1974 and 289,519 caused the entire string of 10 digits (plus commas) to be recognized as a single number. While the number of errors of this type was small, their effect would have been outsized because of the high volumes that resulted.

While it was technically possible to modify the number recognition rules to address such cases, such changes would have had to be made in the “tokenizer” module, which is one of the first modules to run in the process and one on which all other processing depends. The validation was performed late in the development process, so the risk of such fundamental changes was very high. Given that the validation rules could easily identify such cases, the decision was made to edit the text versions of the source files to add the missing spaces.

The analysis of small weights and volumes identified several types of issues, although again many of them were related to optical character recognition errors. The most common error was the insertion of a period instead of a comma. In the example shown in Figure 2, the highlighted number was transcribed as 1.217,331, with a decimal point in the second position, rather than 1,217,331.



\$1,726,425. The District used Manson Construction & Engineering Company during the second quarter of FY 00 to dredge the Bar Channel using the dredges "Newport" and "Bayport". The dredges removed 1.217,331 cubic yards of material at a cost of \$2,759,772. The District prepared Plans and Specifications for the FY 00 Brunswick and Savannah Entrance Channel, Station 50+000 to 112+500, and the Sediment Basin and Station 0+000 to 50+000 contracts.

Figure 2: Example of Source Text for OCR error

The source PDF file for each example of this type was examined, and if the correct value could be determined, the extracted text was edited to correct the error. In one case, such correction was not possible. In the example shown in Figure 3, the highlighted number was interpreted as 2,078,221. From the source text, it was not possible to determine what the correct value of the volume should be. In such cases, the data was left as is.

Agitation dredging: Four pump barges were operated in the Vicksburg district moving a total of 3,863,504 cubic yards of material (2,078,221 cubic yards at Duckport to Delta Point (598-602)¹ 1,538,688 cubic yards at Racetrack Towhead (606-612), and 246,595 cubic yards at Shipland Point (548), at a total cost of \$296,071.78.

Figure 3: Example of Uncorrectable Error in Source Text

The check for weights and volumes with leading zeros identified OCR errors of a different type. Many of the source documents were clearly scanned from bound volumes. As such, the text at the edges of the pages curves where the paper could not be pressed tightly against the glass of the scanner. The optical character recognition package did not account for the curvature and attempted to identify text by reading along straight lines across the page. In some cases, this simply reduced the accuracy of the recognition process.

Tacoma, Grays Harbor, and Olympia, Wash., revised 1963.)
Operations and results during fiscal year. Regular funds:
Maintenance, hired labor: Channel condition surveys were made.
U.S. hopper dredge *Pacific*, October 5, 1965, to March 7, 1966,
removed 488,415 cubic yards from Sand Island shoal, and 385,
075 cubic yards from Crossover channel, a total of 873,490 cubic
yards, bin measurement. Maintenance, contract: Pipeline
dredge *Robert Gray*, leased from Port of Grays Harbor Commis-
sion, removed 790,690 cubic yards of material from Chehalis
River and north channel, July to December; and 30,000 cubic
yards

Figure 4: Example of impact of Binding Curve in Scanned Text

An example of the impact of this curvature is shown in Figure 4. The recognized text for the lines that include the highlighted numbers was

removed 488,415 cubic yards from Sand Island shoal, and 38bi
075 cubic yards from Crossover channel, a total of 873,490 cu:

The performance of the OCR was clearly degraded in the curved region and the result was that the value of “075 cubic yards”, rather than “385,075 cubic yards,” was extracted. There is no way to account for such errors as part of the natural language processing, so when such situations were identified the source text was corrected.

The summation of weights and volumes and comparison of values across individual decades was subject to annual variations in the amount of dredging work that was performed. As shown in Figure 5, across the entire span of time, the differences in volumes was considerable. However, the changes within each decade were generally relatively small. The most significant exception was the 1940s, where the historical events likely resulted in changes in priorities and a dramatic change in dredging between the first half and the second half of the decade. Even then, the range within the year was within the factor of ten defined as the threshold. The drop in volume between 1944 and 1945 was specifically investigated, as described below in connection with the number of removal operations.

The graph in Figure 5 suggested an alternate metric that could be used for identifying potential problems: the year to year variation. In particular, if there were problems extracting data from a particular file, it could manifest itself in a sudden drop in counts and volumes. Looking at Figure 5, one can see that after the change between 1944 and 1945, the next most significant year-to-year drop is between 1965 and 1966. No particular explanation comes to mind for this drop, which warranted further investigation.

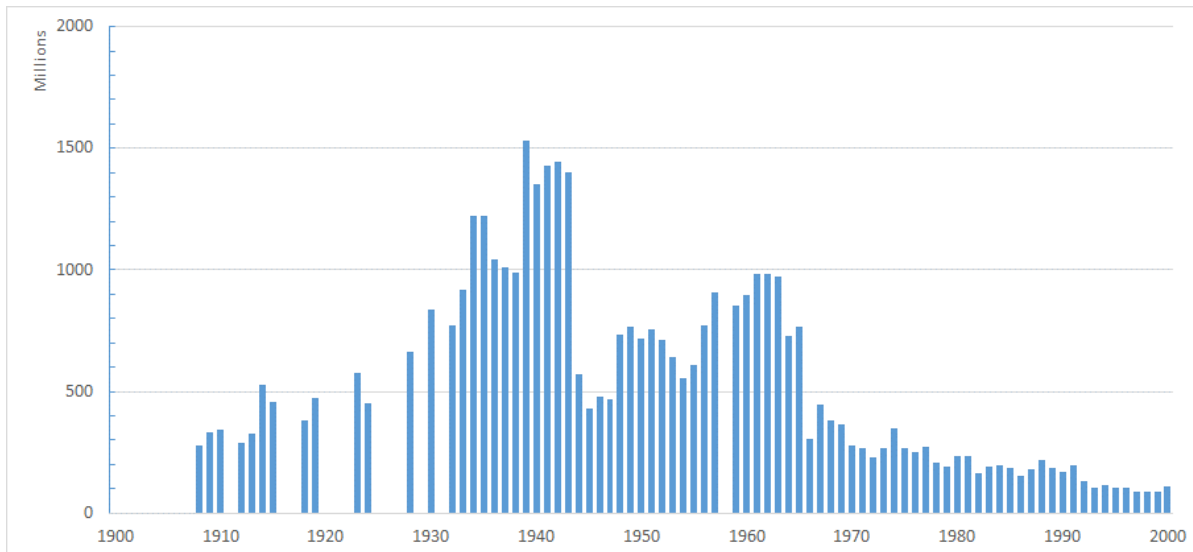


Figure 5: Initial Sum of Removal Volumes per Year (cubic yards)

It was noted that the file for 1966 was one that had a significant number of pages with curved text at the edges, as described above. It soon became apparent that the curvature was causing greater problems than an occasional recognition error. An example of the degree of problems caused by the curvature is shown in Figure 6 and Figure 7.

are considered adequate for existing commerce. Hopper dredge
Operations and results during fiscal year. Hopper dredge
Essayons and attendant plant were employed intermittently from
July 12 to August 1, 1965, in dredging to restore sections of Bay
Ridge and Red Hook Channels to a partial depth of 35 feet. Re-
moved 204,200 cubic yards, place measurement, of material at a
total cost of \$168,353, including \$1,389 for engineering prelimi-
nary to dredging for maintenance.

Figure 6: Sample extract from 1966 Chief's Report, with extreme text curvature

Operations and results during commerce.
Essayons and attendant plant were fiscal year. Hopper dredge
July 12 to August 1, 1965. in dredging, employed intermittently fr,"
 Ridge and Red Hook Channels to a partial to restore sections of BAY
 depth of 35 feet. ge.
 moved 204,200 cubic yards, place
 total cost of \$168,353, including measurement, of material at
 \$10or en
 nary to dredging for maintenance.

Figure 7: Text extracted from the paragraph shown in Figure 6

One can see that the curvature, in addition to reducing the quality of the recognition process near the edges, causes the text extraction process to produce text that is not in the intended order. (The red lines in Figure 7 illustrate how the text should read.) To assess the level of impact of these errors, the text version of the file was manually edited to correct the content. Given the size of the file (over 1800 pages), a complete manual transcription was not realistic. Instead, the process focused on the text describing removal and placement operations. In particular, the procedure involved searching for instances of the string “cubi”, as in “cubic yards” that would be found in nearly all volume references. When this string was found in the “Operations during current fiscal year” section, the text was updated to match that found in the corresponding section of the PDF version of the document. Applying this process to the entire file resulted in a dramatic change in the extracted information, as shown in Table 1. The number of identified operations went up by over 30% and the volume of material removed increased by over 60%.

Table 1: Effect of Correcting OCR Errors due to Text Curvature

	Removal Operations	Removed Volume
Original File	313	286,427,674
Edited File	408	548,063,486
Percent Change	30.4%	61.0%

A survey of the documents from that era revealed that the files from 1965 through 1969 were all scanned from bound volumes. The degree of curvature observed in 1965, 1968 and 1969 was noticeably less than 1966 and 1967. Given limited resources and the significant time required to perform the manual corrections to the document, only the 1966 and 1967 files were edited to correct the curvature-induced errors. Figure 8 shows the sum of volumes after applying these corrections, where it is clear that the transitions in the 1965-1966 period are much smoother.

The counts of the number of placement and removal operations per year followed a similar pattern as the summation of weights and volumes, in terms of temporal variation. In fact, the counts varied more widely. The counts of removal operations over the 1940s had a coefficient of variation of 28.3%, exceeding the threshold. This might be explainable by the historic activities at the time, and manual examination of the recall of the 1945 file, which had the lowest number of removal operations for the decade, indicated that the recall

exceeded 90%. This suggests that the change in the number of operations was real, rather than the result of a problem with extraction process when applied to that file. This operation-count measure turned out to be a very poor metric for placement operations, where five of the eleven decades failed the test. It appears that placement operations, which are less numerous in every year than removal operations, are reported with less regularity as well. This was reflected in the volume-per-year metric for placement operations as well, where three of the eleven decades failed the test.

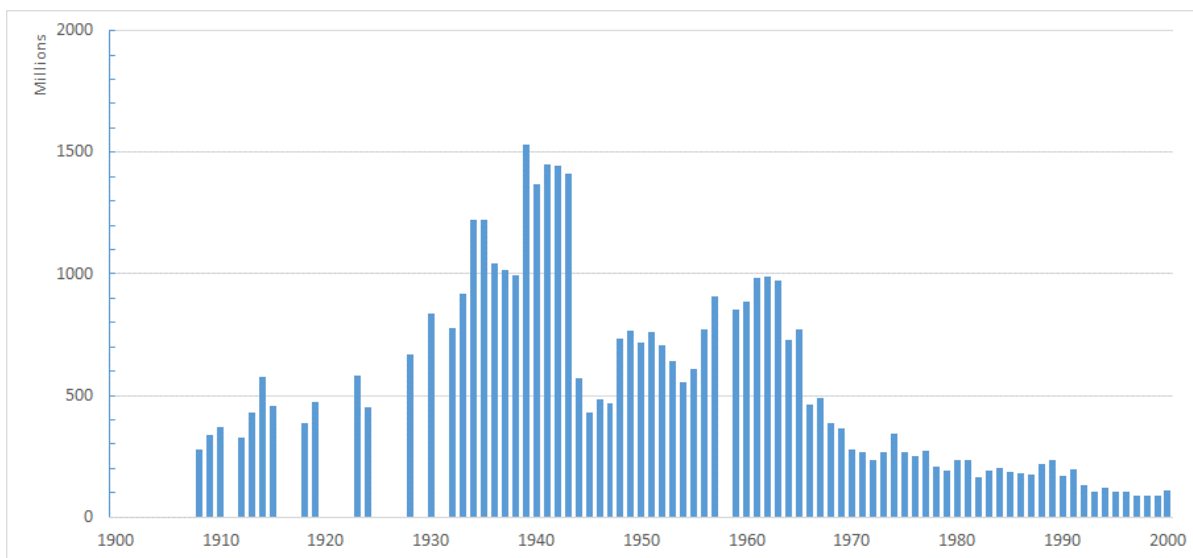


Figure 8: Revised Sum of Removal Volumes per Year (cubic yards)

In summary, the final delivered data passed the vast majority of the validation test. The only exceptions, as explained above, were the removal counts during the 1940s and the placement volume and counts during several decades. In each case where the tests failed, manual inspection revealed no apparent problems with the extraction process.

4.1.3 Validation results (Version 2.0)

The validation rules described in section 4.1.1 were applied to the placement and removal records extracted from the entire corpus. Specific records or input documents that were identified by these rules were then investigated manually to determine if there was any errors or issues. When such errors were identified, suitable corrections were applied where possible.

Note that the Version 2 processing included handling of tabular data in the Galveston District from 1968 through 2000. This processing required manual edits to the text version of the Chief's reports to reorder the data for accurate processing. Given the potential for errors in this manual step, the results processing of the Galveston data were reviewed for coverage during each of the affected fiscal years.

The processing for invalid number formats and high and low number ranges did not uncover any systematic errors but did result in correction of some OCR-induced errors in the input documents.

The results of the comparison of the total volume extracted are shown in Figure 9. The year-to-year transitions in the graph are generally smooth. Note that the drop during the 1940s that is present in is replaced with a more gradual decline.

Note that the values for some years in Figure 9 are a bit larger than those shown in Figure 8. This demonstrates the improved recall of the revised process. Note also that this increase is in spite of the elimination of the double counting associated with the sum-constituent processing. The Constituent/Total processing identified over a total of 2 billion cubic yards of double counting in the Removal operations.

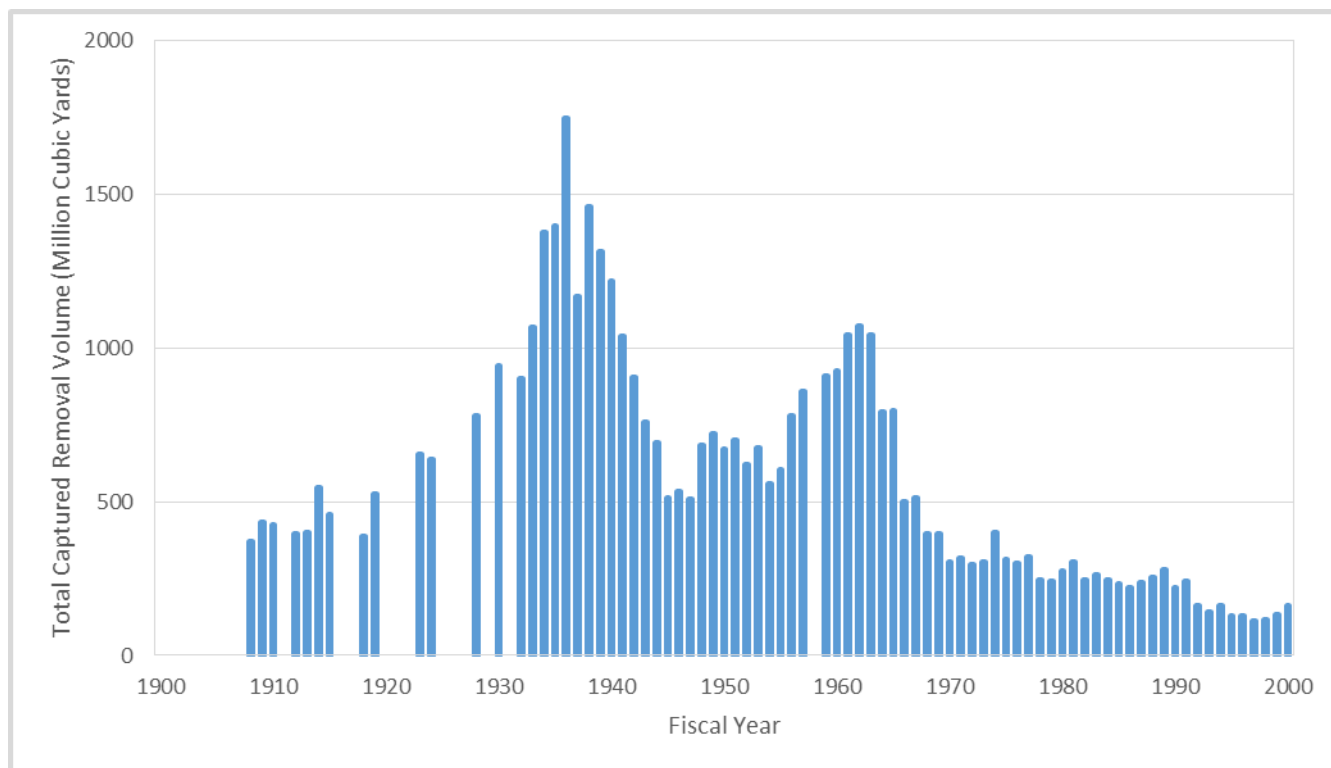


Figure 9: Version 2 Sum of Removal Volumes per Year (cubic yards)

The Fiscal year coverage analysis focused on coastal districts, as was recommended by the ACE SMEs. The checking was an iterative process, where the input files and process rules were examined in cases where gaps were identified, and suitable corrections made. The initial intention was to look for large temporal gaps. After those were resolved, some single-year gaps become very noticeable, particularly if there were a large number of Removal operations during the years before and after. Such instances were also examined and identified issues were corrected. The results for the districts included in this analysis are described in detail below. The overall result is that no any remaining gaps in the data were verified to be true, either as part of this validation operation or as part of the ACE SME effort

that produced the Missing Data report. In two districts with a significant number of gaps (Honolulu and Huntington), the number of Removal operations when detected was small, so having years with none at all is reasonable.

- Alaska: Only a few entries prior to 1950. Removal operations captured for all years between 1950 and 2000 except 1954. Alaska district was chartered in 1949. Manually confirmed that there is only a single Placement operation reported in 1954. Removal operations were captured in 1923, 1928, 1930 and 1932 for the Juneau, Alaska District.
- Baltimore: Data for every year between 1908 and 2000 except 1973. Manual inspection identified only one possible removal operation in 1973, and original text was ambiguous.
- Charleston: Removal operations captured for all years between 1950 and 2000 except 1915. Manually verified that there were no reported removal operations in 1915.
- Galveston: Data for every year between 1908 and 2000 except 1964. Manually verified that no removal operations were reported in 1964.
- Honolulu: Removal operations found in some years, but not in others. In years with reported operations, the number is small (always <10, often 3 or fewer), so it is possible that there were no operations during some years. Removal operations were captured in 1933 and 1936, two years that were cited as gaps in the Missing Data report.
- Huntington: Removal operations found in some years, but not in others. In years with events, the number is small (always <5), so it is possible that there were no operations during some years. Removal operations were captured in 1975, a year that was cited as a gap in the Missing Data report.
- Jacksonville: Removal operations were captured for every year between 1909 and 1980 except 1918. As noted in Missing Data report, after 1980 the reports contained no data or cost only. Manually verified that there was no removal data 1918.
- Los Angeles: Removal operations were captured for most years between 1908 and 1981. A small number of removal operations were identified in few years during the 1990s. No removal operations were identified in 1935, 1964, 1965 and 1967. Manual inspection showed several placement operations but no removal operations in those years. The Missing Data report noted that after 1981, reports indicated dredging was performed, but without specifying quantity or cost.
- Mobile: Removal operations were captured for every year between 1908 and 1991 and between 1994 and 1996. Manually verified that there was no data in 1992 and 1993.
- New England: Removal operations were captured for every year between 1908 and 2000.
- New Orleans: Removal operations were captured for every year between 1908 and 2000.
- New York: Removal operations were captured for every year between 1908 and 2000.

- Norfolk: Removal operations were captured for 1909 and every year between 1913 and 2000. This district was not mentioned in the Missing Data report. Manually confirmed there were no removal operations in 1910 and 1912.
- Philadelphia: Removal operations were captured for every year between 1908 and 2000.
- Portland: Removal operations were captured for every year between 1908 and 2000.
- Sacramento: Removal operations were captured for every year between 1930 and 1964. Sacramento district was formed in 1930. As noted in Missing Data report sampling, after 1964 the reports contained no data or cost only.
- San Francisco: Removal operations were captured for 1908, 1910, 1914 and most years between 1918 and 2000, except 1984-1991. This district was not mentioned in the Missing Data report. Manual sampling confirmed that there were no removal operations reported in 1984, 1986, and 1987.
- Savannah: Removal operations were captured for every year between 1908 and 2000.
- Seattle: Removal operations were captured for every year between 1908 and 1991 except 1944. Data are sporadic after 1991.
 - 1944: Some placement data present in the document. Found two removal operation listed in table form that existing rules were not designed to handle:
 - Project: PUGET SOUND AND ITS TRIBUTARY WATERS, WASH.
 - Navigation Path: Skagit River
 - Removal: 27,145 cubic yards of sand
 - Project: PUGET SOUND AND ITS TRIBUTARY WATERS, WASH.
 - Navigation Path: Skagit River
 - Removal: 10 cubic yards of rock
- Vicksburg: Removal operations were captured for every year between 1908-1912 and 1923-1991 except 1946. This district was not mentioned in the Missing Data report. Manually verified that there were no removal operations in 1946.
- Walla Walla: Removal operations captured sporadically between 1950 and 1972. The Missing Data report indicated that district was formed in 1948 and that later files had no data.
- Wilmington: Removal operations were captured for every year between 1908 and 2000.

5. Data Delivery Format

The natural language information extraction tool that was used in this project is not a database and thus does not readily produce the relational tables that would fully represent the structured information as represented by the data model. The tool was adapted to produce files that incorporate data from multiple tables and to incorporate key and reference information that will enable the population of database tables. This section describes the file formats and a proposed database schema for representing the data. It also presents a proposed database schema that can be used to store and process the resulting data.

5.1 Data File Format

The data produced by the extraction operation is broken into three separate files, focusing on the data that is central to the process: the placement activities, the removal activities and the operations that unite them. The data from the other parts of the model are integrated with the data from these entities into the three files. As shown in Figure 10, the *Operation* file contains information from the ACE_Operation and ACE_Date entities. The *Placement* and *Removal* files contain data from their respective entities, along with data from the ACE_District, ACE_Project and a portion of the ACE_NavigationPath entities. All of the files contain the fiscal year information from the ACE_Document entity. In addition, the files have index information that enables cross-referencing of entities from one file to another.

The files are in comma-separated format. They each include a header row that specifies the values of the columns.

Note: The ID values that are used as indices for cross-referencing entities across documents are 16-17 digit integers. These numbers use the annotation ID produced by the GATE natural language processing tool for each instance as a starting point. The IDs generated by GATE are only unique among annotations produced from within the same file. (i.e., all the annotations produced by processing a single Chief's report document would be unique.) To ensure that the values are unique across all documents, a hash code generated from the source file name was prepended onto the annotation ID.

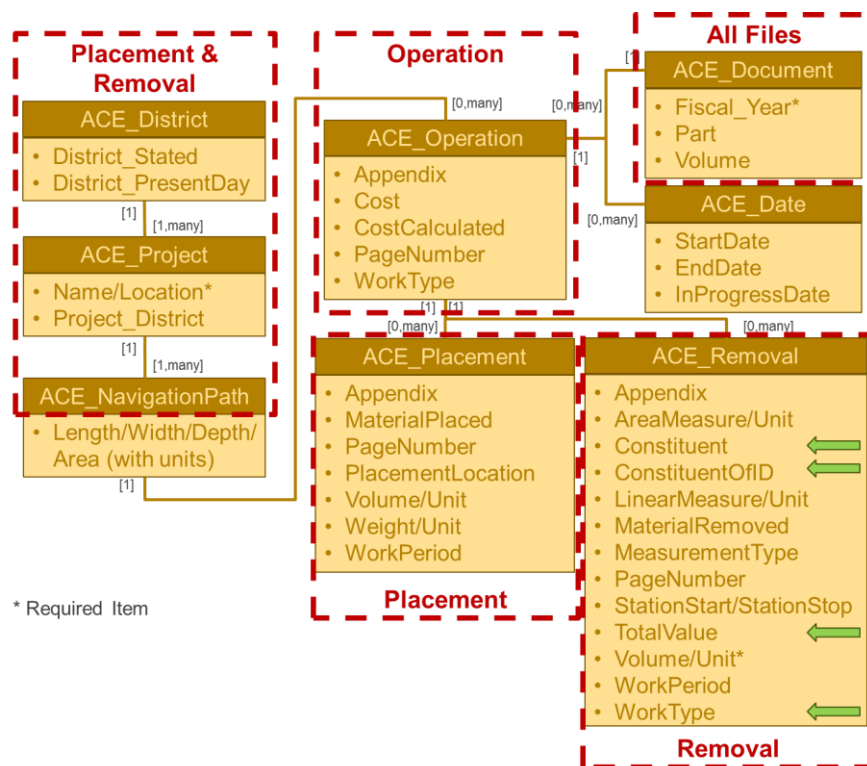


Figure 10: Mapping of Data Model to Data Files

5.1.1 Operation File

The Operation file captures information from the ACE_Operation entity, along with data from the ACE_Date entity. Each line in this file corresponds to one instance of the ACE_Operation entity and one, two or three instances of the ACE_Date entity.

The following elements are included in the Operation file:

- Elements associated with the ACE_Operation entity
 - ACE_Operation attributes: Abstract, Cost, CostCalculated, PageNumber
 - OperationID: primary key for this instance of the ACE_Operation entity
- Attributes from the ACE_Document entity
 - FiscalYear: date of the file
 - FileVolume: volume number/numeral of the document
 - FilePart: part number/numeral of the document
- Attributes from the ACE_Date entity
 - StartDate1/EndDate1: starting and ending date from the ACE_Date entity. Could populate one, the other or both. If present, represents an instance of the ACE_Date entity.

- StartDate2/ EndDate2: starting and ending date from the ACE_Date entity. Could populate one, the other or both. If present, represents an instance of the ACE_Date entity.
- InProgressDate: in-progress date from the ACE_Date entity. If present, represents an instance of the ACE_Date entity.
- Miscellaneous elements
 - Type: set to “Operation” for every record
 - Offset: Count of characters, from the start of the text file to starting position of the annotation. Not part of the data model, used for debugging purposes.

The order of the attributes is:

Type, FiscalYear, OperationID, StartDate1, EndDate1, StartDate2, EndDate2, InProgressDate, Cost, CostCalculated, PageNumber, Abstract, Offset, FileVolume, FilePart

5.1.2 Placement File

The Placement file captures information from the ACE_Placement entity, along with data from the ACE_District, ACE_Project and ACE_NavigationPath entities. Each line in this file corresponds to one instance of the ACE_Placement entity. Each line references one instance of each of the ACE_District, ACE_Project and ACE_NavigationPath entities. More than one line in the file can (and will) point to each of those entities. (i.e., there will likely be more than one placement operation in each given district each year.)

The following elements are included in the Placement file:

- Attributes from the ACE_Document entity
 - FiscalYear: date of the file
 - FileVolume: volume number/numeral of the document
 - FilePart: part number/numeral of the document
- Elements associated with the ACE_Operation entity
 - OperationID: cross reference (foreign key) to corresponding ACE_Operation record
- Elements associated with the ACE_District entity
 - ACE_District attributes: StatedDistrict, PresentDayDistrict
 - DistrictID: cross reference (foreign key) to corresponding ACE_District record
- Elements associated with the ACE_Project entity
 - ACE_Project attributes: ProjectDistrict, ProjectName

- ProjectID: cross reference (foreign key) to corresponding ACE_Project record
- Attributes associated with the ACE_Placement entity
 - Abstract, ApproxPageNumber, MaterialPlaced, MeasurementType, PlacementLocation, RemovedFromNavPath, VolumePlaced, VolumePlacedUnit, WeightPlacedUnit, WeightPlaced, WorkPeriod, WorkType
- Miscellaneous elements
 - Type: set to “Placement” for every record
 - Offset: Count of characters, from the start of the text file to starting position of the annotation. Not part of the data model, used for debugging purposes.

The order of the attributes in the file is:

Type, FiscalYear, Work Period, StatedDistrict, PresentDayDistrict, ProjectDistrict, ProjectName, PlacementLocation, RemovedFromNavPath, VolumePlaced, VolumePlacedUnit, MeasurementType, WeightPlaced, WeightPlacedUnit, MaterialPlaced, WorkType, OperationID, DistrictID, ProjectID, ApproxPageNumber, Abstract, Offset, FileVolume, FilePart

5.1.3 Removal File

The Removal file captures information from the ACE_Removal entity, along with data from the ACE_District, ACE_Project and ACE_NavigationPath entities. Each line in this file corresponds to one instance of the ACE_Removal entity. Each line references one instance of each of the ACE_District, ACE_Project and ACE_NavigationPath entities. More than one line in the file can (and will) point to each of those entities. (i.e., there will likely be more than one removal operation in each given district each year.)

The following elements are included in the Removal file:

- Attributes from the ACE_Document entity
 - FiscalYear: date of the file
 - FileVolume: volume number/numeral of the document
 - FilePart: part number/numeral of the document
- Elements associated with the ACE_Operation entity
 - OperationID: cross reference (foreign key) to corresponding ACE_Operation record
- Elements associated with the ACE_District entity
 - ACE_District attributes: StatedDistrict, PresentDayDistrict
 - DistrictID: cross reference (foreign key) to corresponding ACE_District record

- Elements associated with the ACE_Project entity
 - ACE_Project attributes: ProjectDistrict, ProjectName
 - ProjectID: cross reference (foreign key) to corresponding ACE_Project record
- Attributes associated with the ACE_Removal entity
 - Appendix, ApproxPageNumber, AreaMeasure, AreaMeasureUnit, LinearMeasure, LinearMeasureUnit, MaterialRemoved, MeasurementType, RemovedFromNavPath, StationStart, StationStop, VolumeRemoved, VolumeRemovedUnit, WorkPeriod, WorkType,
- Miscellaneous elements
 - Type: set to “Placement” for every record
 - Offset: Count of characters, from the start of the text file to starting position of the annotation. Not part of the data model, used for debugging purposes.

The order of the attributes in the file is:

Type, FiscalYear, Work Period, StatedDistrict, PresentDayDistrict, ProjectDistrict, ProjectName, RemovedFromNavPath, VolumeRemoved, VolumeRemovedUnit, MeasurementType, MaterialRemoved, LinearMeasure, LinearMeasureUnit, AreaMeasure, AreaMeasureUnit, StationStart, StationStop, WorkType, TotalValue, Constituent, ConstituentOfID, OperationID, DistrictID, ProjectID, ApproxPageNumber, Appendix, Offset, FileVolume, FilePart

5.2 Proposed Database Schema

A proposed schema for representing this information in a database is shown in Figure 11. In general, the schema aligns with the data model, with most of the data model entities mapping to individual tables. There are tables corresponding to the ACE_Date, ACE_District, ACE_Document, ACE_Placement, ACE_Project, ACE_Operation and ACE_Removal entities. Each of these tables has columns that correspond to each of the attributes defined for the corresponding entity.

(ADDED) Note: The proposed schema was NOT updated to reflect the fields that were added to the Removal Table. Extension of the schema to include the two new binary values (TotalValue and Constituent) and the one foreign key (ConstituentOfID) are straightforward.

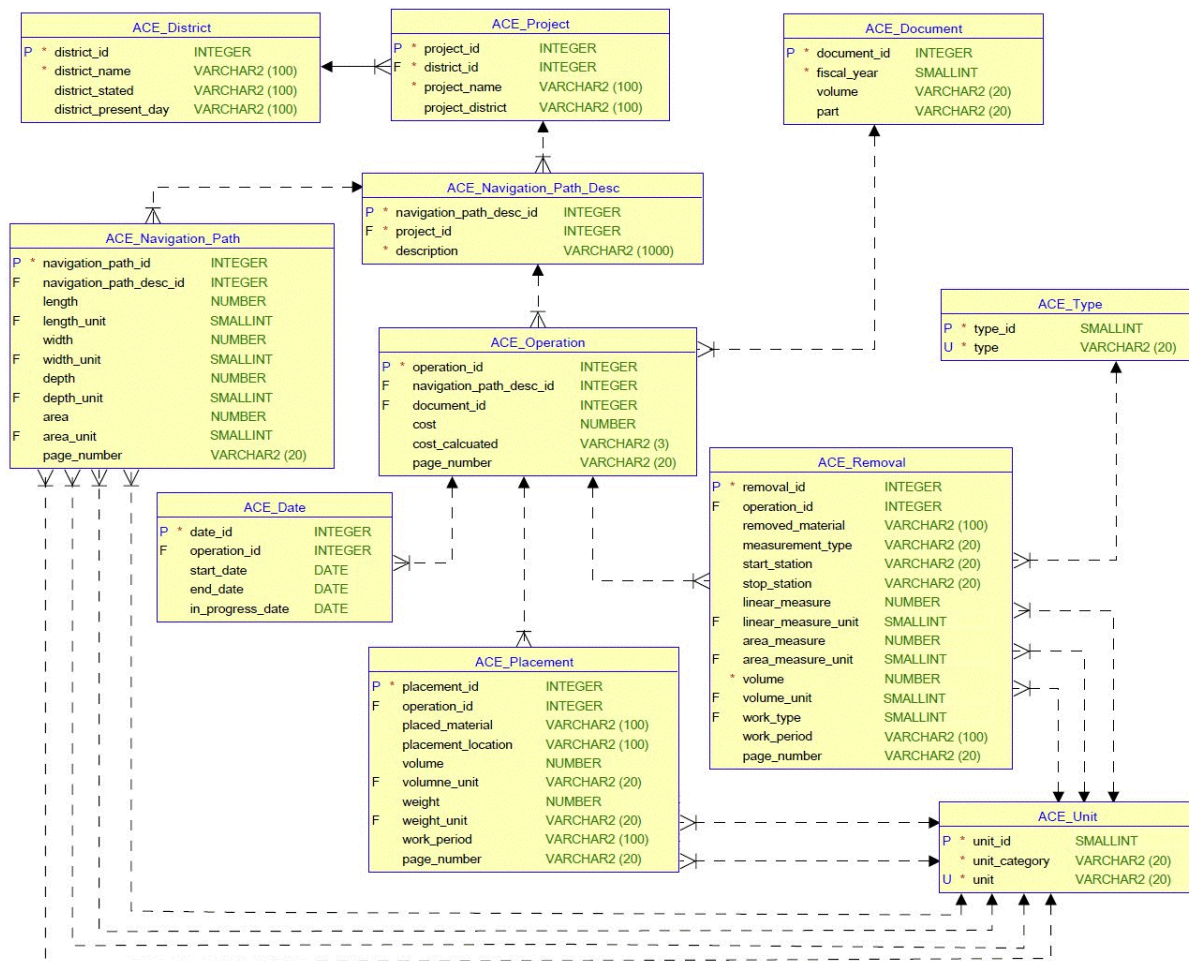


Figure 11: Proposed Database Schema for Extracted Information

The schema breaks the ACE_NavigationPath entity into two tables. The rationale for this is the two purposes for the navigation path concept – to capture the physical dimensions and characteristics and to represent the organizational hierarchy of each removal and placement operation. The ACE_Navigation_Path_Desc table is used to represent the hierarchy and the ACE_Navigation_Path table is used to capture the physical characteristics.

The table structure also creates some reference-data tables, used to represent values that only have a small number of allowed values. These include:

- ACE_Type: This table stores the three types of work that can characterize the operations: “Environmental”, “New Work”, and “Maintenance”
- ACE_Unit: This table stores the various units that can represent the dimensions of navigation paths and material that is removed or placed: “acres”, “cubic yards”, “feet”, “foot”, “linear feet”, “miles”, “pounds”, “square feet”, “square miles”, “square yards”. “tons”

The following details the columns within the individual tables

ACE_Date

- date_id: primary key of an entry in this table (INTEGER, required)
- operation_id: foreign key, pointing to entry in ACE_Operation table (INTEGER)
- start_date: date on which operation, or phase of operation, began (DATE)
- end_date: date on which operation, or phase of operation, finished (DATE)
- in_progress_date: date that operation was reported to be underway (DATE)

ACE_District

- district_id: primary key of an entry in this table (INTEGER, required)
- district_name: string representing the name of the district. (VARCHAR2(100), required)
- district_stated: string representing district as declared in the document (VARCHAR2(100))
- district_present_day: string representing district that currently manages the associated projects (VARCHAR2(100))

ACE_Document

- document_id: primary key of an entry in this table (INTEGER, required)
- fiscal_year: four digit representation of year of document (SMALLINT, required)
 - Note: This cannot be unique per document because in some cases the descriptions of operations span multiple years.
- volume: volume of the document, as reported in the text (VARCHAR2(20))
 - Note: this cannot be treated as an integer because in some cases the volume numbers are reported using Roman numerals
- part: part of the document, as reported in the text (VARCHAR2(20))
 - Note: this cannot be treated as an integer because in some cases the part numbers are reported using Roman numerals

ACE_Navigation_Path_Desc

- navigation_path_desc_id: primary key of an entry in this table (INTEGER, required)
- project_id: foreign key, pointing to entry in ACE_Project table (INTEGER, required)
- description: textual name of the project, taken from document (VARCHAR2(1000), required)

ACE_Navigation_Path

- navigation_path_id: primary key of an entry in this table (INTEGER, required)
- navigation_path_desc_id: foreign key, pointing to entry in ACE_Navigation_Path_Desc table (INTEGER)
- area: coverage measure of navigation path (NUMBER)
- area_unit: unit of area measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- depth: downward measure of navigation path (NUMBER)
- depth_unit: unit of depth measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- length: end-to-end distance measure of navigation path (NUMBER)

- length_unit: unit of length measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- width: side-to-side distance measure of navigation path (NUMBER)
- width_unit: unit of width measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- page_number: page in document where information was found (VARCHAR(20))
 - Note: this is defined as a string, to allow for alternate numbering formats, such as SECTION-PAGE.

ACE_Operation

- operation_id: primary key of an entry in this table (INTEGER, required)
- navigation_path_desc_id: foreign key, pointing to entry in ACE_Navigation_Path_Desc table (INTEGER)
- document_id: foreign key, pointing to entry in ACE_Document table (INTEGER)
- cost: dollar value associated with the project (NUMBER)
 - Note: extracted cost values include the dollar sign (e.g., \$235,323) which must be stripped to store these values as numbers.
- cost_calculated: indication of whether cost entry is a sum of values reported in the document (yes/no, VARCHAR(2))
 - Note: in extracted text, this is present and set to “yes” when true, omitted otherwise.
- page_number: page in document where information was found (VARCHAR(20))
 - Note: this is defined as a string, to allow for alternate numbering formats, such as SECTION-PAGE.

ACE_Placement

- placement_id: primary key of an entry in this table (INTEGER, required)
- operation_id: foreign key, pointing to entry in ACE_Operation table (INTEGER, required)
- placed_material: string describing the substance that was placed. (VARCHAR2(100))
- placement_location: string describing destination of placement operation (VARCHAR2(100))
- volume: quantity of material that was placed, in terms of occupied space (NUMBER)
- volume_unit: unit of volume measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- weight: quantity of material that was placed, in terms of mass (NUMBER)
- weight_unit: unit of weight measurement, foreign key pointing to an entry in ACE_Unit table (SMALLINT)
- work_period: description of when work took place, as referenced in the document (e.g., During current fiscal year). (VARCHAR2(100))
- page_number: page in document where information was found (VARCHAR(20))
 - Note: this is defined as a string, to allow for alternate numbering formats, such as SECTION-PAGE.

ACE_Project

- project_id: primary key of an entry in this table (INTEGER, required)
- district_id: foreign key, pointing to entry in ACE_District table (INTEGER, required)
- project_name: string representing the name of the project. (VARCHAR2(100), required)
- project_district: string representing the name of the district that currently manages the project (VARCHAR2(100))

ACE_Removal

- removal_id: primary key of an entry in this table (INTEGER, required)
- operation_id: foreign key, pointing to entry in ACE_Operation table (INTEGER, required)
- removed_material: string describing substance that was removed (VARCHAR2(100))
- measurement_type: string describing the means by which the volume of material was determined (VARCHAR2(20))
- start_station: starting point of work, referenced to navigation path markers (VARCHAR2(20))
- stop_station: ending point of work, referenced to navigation path markers (VARCHAR2(20))
- linear_measure: length along navigation path where removal operation took place (NUMBER)
- linear_measure_unit: unit of length measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- area_measure: size of region where removal operation took place (NUMBER)
- area_measure_unit: unit of area measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- volume: quantity of material that was removed, in terms of occupied space (NUMBER)
- volume_unit: unit of volume measurement, foreign key pointing to entry in ACE_Unit table (SMALLINT)
- work_type: description that characterizes rationale for work, foreign key pointing to entry in ACE_Type table (SMALLINT)
- work_period: description of when work took place, as referenced in the document (e.g., During current fiscal year) (VARCHAR2(100))
- page_number: page in document where information was found (VARCHAR(20))
 - Note: this is defined as a string, to allow for alternate numbering formats, such as SECTION-PAGE.

ACE_Type

- type_id: primary key of an entry in this table (INTEGER, required)
- type: string describing the type of work (“Environmental”, “New Work”, and “Maintenance”) (VARCHAR2(20), required, unique)

ACE_Unit

- unit_id: primary key of an entry in this table (INTEGER, required)

- unit_category: string describing type of measurement metric captures (e.g., volume, area) (VARCHAR2(20), required, unique)
- unit: string describing the measurement metric (VARCHAR2(20), required, unique)

Note that the data provided in the extraction files will need to be adjusted, segregated and merged in order to populate this schema. In particular, the data for the ACE_District and ACE_Project tables will need to be separated from the removal and placement information, and the data for the ACE_Date table will need to be separated from the operation information. The linkages indices provided in the tables provide the basis for generating the foreign keys that link the tables together. Finally, the date information will need to be parsed to populate a DATE type that can be used for subsequent queries.

6. Information Extraction Design Details

The following sections describe the flow of rules that are used to perform the entity extraction processing for the Chief’s Annual Reports. This information documents the process in detail. Those not interested in such detail can skip this section.

The process made use of the General Architecture for Text Engineering (GATE), an open-source tool produced by the University of Sheffield.² The GATE tool provides the means for defining rules, using the Java Annotation Patterns Engine (JAPE) language, that enables concepts taken from Regular Expressions to be applied to Annotations in documents, rather than just to alphanumeric characters. The GATE site includes tutorial information on both the operation of GATE and its various plugins as well as on the JAPE language.

6.1 Document Format and Information

There are elements in the document that need to be processed prior to identifying the named information elements. Some of these are basic items that are identified by the default GATE rules, such as Dates. These will not be covered here. There are some basic elements that are tagged using custom rules to be used in multiple latter stages. These are described here. Finally, in other cases, elements in the structure of the document must be adjusted to enhance the accuracy of the process. These are also described.

6.1.1 Document Date

The document date is a key reference point for identifying the document. The FiscalYear phase identifies this information.

The process works in a single phase.

1. The rules in ACE_fiscal_year.jape look for the first dates in a particular context within the document. If the discovered date is more than a year, the four-digit year is extracted and tagged.
 - a. Input: Token Date
 - b. Logic: Look for a year followed by “ANNUAL REPORT”, preceded by “FISCAL YEAR” or in other constructs that are characteristic of the date information on the first page of the report. Mark the date as an ACE_FiscalYear annotation.

² GATE, <https://gate.ac.uk/>

6.1.2 Document Volume and Part Numbers

A second key set of reference points for identifying the document is the Volume and/or Part number. The DocumentVolume phase identifies this information.

The process works in two phases.

1. The rules in ACE_document_volume.jape look for the first instances in the document that match a pattern that corresponds to the document volume information.
 - a. Input: Token Number
 - b. Logic: “Volume”, “Vol.” or in other constructs followed by a number that are characteristic of the volume information on the first page of the report. Mark the number portion as an ACE_DocVolume annotation.
2. The rules in ACE_document_part.jape look for the first instances in the document that match a pattern that corresponds to the document part information.
 - a. Input: Token Number
 - b. Logic: “Part.” or in other constructs followed by a number or Roman numeral that are characteristic of the part information on the first page of the report. Mark the number portion as an ACE_DocPart annotation.

6.1.3 Line Feeds and Multi-Spaces

In many cases, the identification of items of interest depends on the structure of the document. There particular elements of that structure are line feeds, form feeds and multiple consecutive spaces. This process will create special annotations for these constructs so that they can be easily recognized in later processing.

The process works in a single phase.

1. The rules in ACE_linefeed_multispace.jape look for control space tokens and multiple consecutive space tokens and tags them as LineFeed and MultiSpace, respectively.
 - a. Input: SpaceToken Token
 - b. Logic:
 - i. Look for SpaceToken with kind=control whose text is not a form feed character and annotate as LineFeed.
 - ii. Look for other SpaceToken annotations with kind=control and annotate as FormFeed.
 - iii. Look for multiple consecutive SpaceToken annotations of kind=space and annotate as MultiSpace
 1. Looks for minimum of 2 and maximum of 200 SpaceToken annotations

2. Including Token in input set ensures that rule stops processing when a non-space character is encountered
2. The rules in ACE_fix_form_line_feed.jape removes LineFeed annotations immediately after FormFeed and immediately before an all-caps Token annotation. This will allow the subsequent text to be recognized as a header. This process is limited to post 1968, two-column files.
 - a. Input: FormFeed LineFeed Token
 - b. Logic: Look for consecutive FormFeed and LineFeed annotations followed by an all-caps Token annotation. If the FiscalYear feature of the document is greater than 1967, remove the LineFeed annotation.

6.1.4 Numbers

Numbers play a special role in many of the attributes of interest. The GATE tokenizer identifies numbers, annotating them with a feature of kind=number. In some cases, it is useful to have numbers identified without having to pull in all tokens. In addition, the tokenizer does not accurately capture numbers that contain commas. The number processing rectifies this latter situation and annotates all numbers as Number.

The process works in a single phase.

1. The rules in ACE_number.jape implement this process.
 - a. Input: SpaceToken Token
 - b. Logic: The rules identify tokens where the “kind” feature is set equal to “number”, and for patterns of more than one such token separated by a comma, and annotates the result as Number.

6.1.5 Extraneous Split Removal

In certain cases, the Sentence Splitter function inserts the Split annotation in places that do not represent the end of a sentence. These extraneous Splits can adversely affect rules that depend on the Split annotation as an endpoint for matching. For example, in the following text, the Sentence Splitter would put a Split Annotation after the word DISTRICT, and thus break up the line incorrectly.

RIVERS AND HARBORS--PORTLAND, ME., DISTRICT. 101

The rules in the FixSplit phase correct these case. The process works in a single phase.

1. The rules in ACE_fix_split.jape implement this process.
 - a. Input: Split Token MultiSpace LineFeed
 - b. Logic: The rules identify cases where there is a Split annotation is preceded by or followed by a number

6.1.6 Hyphen Removal

In certain cases, the content of interest is hyphenated, which can cause annotations that rely on specific text to fail to match. For example:

During this period 87,470 cubic yards of ordinary material were removed.

In some cases, the split is across monetary values, such as:

...were removed at a cost of approximately \$67,000.

This processing aims to correct this issue by generating new annotations with string features that omit the hyphen.

The process works in a single phase.

1. The rules in ACE_fix_hyphen.jape implement this process.
 - a. Input: Token LineFeed Money
 - b. Logic: The rules identify cases where there is a Token with a feature “kind = word” followed by a hyphen, LineFeed and another Token with a feature “kind = word”. IN such cases, remove the all the annotations within that extent and add a new Token annotation with a string feature that concatenates the string values of the two original Token annotations, omitting the hyphen and LineFeed. Also Look for Money annotations followed by a hyphen, LineFeed and another Token with a feature “kind = number”. Remove the existing Money annotation and add a new Money annotation over the entire extent that has a TotalCost feature that concatenates the Money annotation and the number, omitting the hyphen and LineFeed.

6.1.7 Other Dividers

In certain cases, it is advantageous to have alternate annotations to represent divisions between phrases or concepts. In particular, commas and semicolons can serve as separators to prevent identification of relationships between separate instances.

The rules in the Dividers phase annotate these elements for use in other phases. The process works in a single phase.

2. The rules in ACE_dividers.jape implement this process.
 - a. Input: Token
 - b. Logic: The rules identify commas and semicolons and in each case generate a Divider annotation.

6.1.8 Page Headers and Page Numbers

The aim of the Page Header processing is to identify page headers in the text and effectively remove them so that they are invisible to any subsequent processing. This accomplished by removing all annotations on the corresponding text, including Token, Split and all other annotation types. Once the annotations are removed, the corresponding text will not be visible to any of the rules that follow. A secondary aim is to identify page numbers in the page headers, or in some cases page footers, and identify them so that annotations can be tagged with the page number where they are found in the source document.

The headers come in two general forms, for left and right-hand pages, such as:

Left: 798 REPORT OF CHIEF OF ENGINEERS, U. S. ARMY, 1941

Right: RIVERS AND HARBORS--NEW ORLEANS, LA., DISTRICT 797

The process is designed to identify headers using general features, rather than being tied to specific ordered sets of words, to allow for differences in headers between documents and even between sections of the document, and to allow for OCR errors may corrupt the text.

The process works in several phases:

1. The rules in ACE_page_header_components.jape look for key words that appear in headers.
 - a. Input: Token
 - b. Logic: Look for single Token annotations or short sets of Token annotations that correspond to specific parts of the header, including “REPORT”, “ENGINEERS”, “U.S. ARMY”, “RIVERS” and “HARBORS”. Annotate these strings as HeaderIndicator.
2. The rules in ACE_page_numbers.jape look for possible page numbers. Need to differentiate these from general numbers because some page headers have other numbers in them.
 - a. Input: Token Number
 - b. Logic: Look for numbers, other than those that are followed by a period or those that have text extensions (e.g., “3RD” or “1st”). Mark those numbers with a PossiblePageNumber annotation.
3. The rules in ACE_page_header_remove.jape find strings that match the structure of the header and removes all contained annotations.
 - a. Input: HeaderIndicator PossiblePageNumber PageNumber DistrictDivision LineFeed Split FormFeed Date
 - b. Logic: Look for patterns of FormFeed, multiple HeaderIndicator/ DistrictDivision and PossiblePageNumber annotations in the sequences that correspond to the structures of the left- and right-hand page headers. The result is marked as a Header annotation. If present, the page number is

- annotated at an ACE_PageNumber, with the text of the page number added as a feature Page. All other annotations are removed from the contained text.
4. The rules in ACE_page_header_remove_generic.jape identify page headers that do not contain any HeaderIndicator or District/Division annotations,
 - a. Input: Token PossiblePageNumber PageNumber LineFeed Split Token FormFeed
 - b. Logic:
 - i. Skip lines of the form “IMPROVEMENT OF RIVERS...”. Look for other constructs that start with the FormFeed annotation, have a PossiblePageNumber in the correct spot (for a left- or right-hand page header) and a set of other unspecified tokens. The result is marked as a Header annotation, with the text of the page number added as a feature where available. All other annotations are removed from the contained text.
 - ii. Look for headers that represent District annotations (starting circa 1970, the District information was put in the page header, rather than in a separate section heading). District annotations have the word “DISTRICT” or “DIVISION” and end with an asterisk. The “Mississippi River Commission” section is identified in the same fashion. Mark those annotations as ACE_District or ACE_Commission, respectively.
 5. The rules in ACE_district_duplicate_removal.jape removes duplicate district annotations found in page headers. This processing is particularly aimed at post-1970s documents where the first mention has an asterisk and subsequent mentions do not.
 - a. Input: ACE_District
 - b. Logic: Look for ACE_District annotation with a rule feature of RightDistrictHeader_First followed by one to 100 ACE_District annotations generated by some other rule. Leave the first ACE_District annotation and remove all the others.
 6. The rules in ACE_page_header_remove_linefeed.jape find headers followed by a line feed and remove the line-feed.
 - c. Input: Header LineFeed Token
 - d. Logic: Look for Header annotation followed immediately by a LineFeed annotation. (Presence of Token in the input set ensures that there is nothing between them.) Remove any annotations contained within the complete extent (including the existing Header annotation), and mark the longer extent with a new Header annotation, copying the features from the original Header annotation.

7. The rules in ACE_page_header_remove_split.jape find headers followed by a split and removes the Split annotation.
 - e. Input: Header LineFeed Split
 - f. Logic: Look for Header annotation followed immediately by a Split annotation. (Presence of Token in the input set ensures that there is nothing between them.) Remove any annotations contained within the complete extent (including the existing Header annotation), and mark the longer extent with a new Header annotation, copying the features from the original Header annotation.
8. The rules in ACE_page_number_pre_header.jape find page numbers ahead of headers, as found in documents starting circa 1970.
 - g. Input: Number LineFeed Header Token Split
 - h. Logic: Look for pattern linefeed-number-linefeed-Header. Verify that the number is strictly digits. If so, increment the number by one, to account for the fact that remaining logic assumes page number is at the top of the page. Annotate the text as ACE_PageNumber, with the string value of the incremented number as a Page feature.
9. The rules in ACE_page_number_overlap.jape one of the PageNumber annotations when there is an overlap.
 - i. Input: ACE_PageNumber
 - j. Logic: Look for ACE_PageNumber annotations contained within ACE_PageNumber annotations. Sort the set of matching annotations by document offset and remove the last one in the list.

6.2 Information Elements

The data model described above contains multiple information elements that the system is designed to identify, including entities, attributes of those entities and relations between them. The rules that have been defined to capture these elements are described in the following sections.

6.2.1 Project

The Chief's Reports documents are organized as a series of sections, each devoted to a particular project. The aim of this processing is to identify the headers within the document that indicate the start of each such section. The sections are numbered and thus have a recognizable pattern:

21. INTRACOASTAL WATERWAY, MERMENTAU RIVER TO CALCASIEU RIVER, LA. (D4)

The processing is accomplished in several phases.

1. The rules in ACE_number_list.jape find lists of Numbers, that might occur in Project headings (e.g., NOS. 1, 2 AND 3) and marks them with a single annotation.
 - a. Input: Token
 - b. Logic: Look for the string “No” followed by a number, with optional additional numbers and the word “and”. Mark the extent as a NumberList annotation.
2. The rules in ACE_project_candidate_headings.jape look for text with this general format.
 - a. Input: Number LineFeed Split Token PossibleProject NumberList
 - b. Logic: Look for a number and period at the beginning of a line, followed by a word in all caps, followed by one or more all-caps or upper-initial Token annotations, with subsequent all-caps, upper initial or punctuation Token annotations, with optional NumberList annotation. The rules allow the header to span at most one LineFeed annotation and allow an optional one parenthetical statement at the end, followed by a terminating LineFeed annotation. The identified string is annotated as ACE_Project. The text that starts with the first all-caps word after the initial number-period, not including any text in parentheses, is normalized (remove multiple consecutive spaces and line feeds) and added as a feature ProjectName. There is also a preceding non-match rule to avoid cases where something that looks like a project name is actually tabular data.
3. The rules in ACE_project_candidate_headings2.jape look for text in the format used for headings in early years, circa 1909.
 - a. Number LineFeed Token Location FullLocation ACE_Project
 - b. Logic: Skip over any existing ACE_Project annotations that occur in a similar context. Then look for in-line project headings of the form “9. Harbor at Boston, Mass.-In its original condition the headlands...”. Annotate the text between the number and the dash as an ACE_Project annotation.
4. The rules in ACE_project_candidate_headings3.jape look for text in the format used for in the Mississippi River Commission section of the documents. These are generally all-caps, possibly with parenthetical material, on one or two lines., such as “BATON ROUGE HARBOR (DEVILS SWAMP), LA.”
 - a. Number LineFeed Split Token
 - b. Logic: Mark text that fits the defined format as a MRCProject annotation.

Note: this pattern can occur in other parts of the document, where it is not a Project heading; subsequent processing will remove it unless it follows an MRC or similar heading.

5. The rules in ACE_project_headings_remove look to remove candidate project headings that do not contain a statement of a location, navigation path or known project.
 - a. This processing depends on several gazetteers:
 - i. ACE_projects--gaz.lst contains a list of existing projects provided by ACE subject matter experts
 - ii. ACE_navigation_path--gaz.lst contains a list of navigation path primitives, such as “anchorage” or “channel”
 - b. Input: ACE_Project FullLocation NavPath PossibleProject
 - c. Logic
 - i. If a candidate project annotation contains a FullLocation, NavPath or PossibleProject annotation, it is left alone
 - ii. Otherwise, the ACE_Project annotation is removed.
6. The rules in ACE_remove_errant_MRCProject remove candidate project headings that are indicated as annotation of type MRCProject that are found before any ACE_Commission annotation in the document.
 - a. Input: ACE_Commission ACE_District MRCProject
 - b. Logic: Locate the first instance of ACE_Commission annotation in the document. Remove any MRCProject annotations that are found prior to that in the document. Stop processing on the first trigger of the rule.
7. The rules in ACE_project_toc_remove.jape remove candidate project headings that are found in the table of contents.
 - a. Predecessor: The rules in ACE_ToC_Indicator.jape identify the table of contents entries by their structure, including the presence of MultiSpace annotations and/or consecutive series of dots. These are marked as ToC annotation.
 - b. Input: ACE_Project ToC Split
 - c. Logic: Any ACE_Project annotation that contains a ToC annotation or is followed immediately by one is removed.
8. The rules in ACE_district_project.jape associate ACE_Project annotations with the ACE_District or ACE_Commission annotations that precede them.
 - a. Input: ACE_Project ACE_District ACE_Commission
 - b. Logic:
 - i. Look for ACE_District annotation followed by an arbitrary number of ACE_Project annotations (currently set to a max of 50). Copy DistrictName, PresentDayDistrict and StatedDistrict features from ACE_District to ACE_Project.

- ii. Look for ACE_Commission annotation followed by an arbitrary number of ACE_Project annotations (currently set to a max of 50). Copy CommissionName feature from ACE_Commission to ACE_Project as ACE_DistrictName.

6.2.2 Navigation Path

Within each site, there are references to navigation paths, such as rivers, channels, and passes. The initial Navigation Path processing looks to identify those paths and to associate them with the corresponding project. Later processing will remove references to those paths for which no dredging information is provided.

The processing is accomplished in multiple phases.

1. The terms in ACE_navigation_path--gaz.lst identify base words for navigation paths, including anchorage, bayou, channel, and river. The gazetteer processing is configured to annotate the matching words as NavPath.
2. The rules in ACE_navigation_path_extend.jape expand the base NavPath annotations identified by the gazetteer processing to include adjacent descriptive words that help uniquely identify each NavPath. These include depth modifiers (e.g., 12-foot channel), location keys (e.g., upper bayou), noun modifiers (e.g., jetty channel) and capitalized words (e.g., Oswego River Basin).
 - a. Input: Token NavPath Lookup Number
 - b. Logic: Look for instances of NavPath annotation preceded or followed by one of the descriptive options. Annotate the extended text as NavPath and remove the NavPath annotation from the original, shorter text.
3. The rules in ACE_navigation_path_merge.jape merges lists of extended navigation paths that are likely treated as a whole (e.g., southwest entrance channel and northern pass).
 - a. Input: Token NavPath
 - b. Logic: Look for pairs or lists of NavPath annotations. Annotate the extended text as NavPath and remove the NavPath annotations from the components included in the list.
4. The rules in ACE_navpath_remove.jape remove NavPath annotations in certain constructs that are not of interest.
 - a. Input: NavPath Token Split
 - b. Logic: Locate NavPath annotations in certain constructs, including “River and Harbors Act” and “construction of a levee along the river”. Remove those NavPath annotations.
5. The rules in ACE_navpath_remove2.jape remove NavPath annotations that are found in tables.

- a. Input: NavPath MultiSpace LineFeed
 - b. Logic: Locate NavPath annotations with multiple spaces on each side or preceded by a LineFeed annotation and a large number of spaces. Remove those NavPath annotations.
6. The rules in ACE_navpath_select.jape retains the longest NavPath reference in a sentence and removes others. For example, in the sentence “Under contract for the removal of shoal areas in the [8-foot channel], between the mouth of the [river] and the cities of Saco and Biddeford”, there are two NavPath annotations marked with square brackets. The NavPath annotation on “8-foot channel” would be retained and the one on “river” would be removed.
- a. Input: NavPath Volume Split
 - b. Logic: Look for two or three NavPath annotations within one sentence, compare their lengths in characters and retain the longest, removing the others. The Volume annotation in the input set serves as another separator, and thus if NavPath annotations in the same sentence have a Volume annotation in between, they will not be compared.
7. The rules in ACE_navpath_project.jape associate detected navigation paths with the project information that precedes them.
- a. Input: NavPath ACE_Project
 - b. Logic: Find ACE_Project annotation and subsequent NavPath annotations, looking for an arbitrary number of the latter, currently set to a maximum of 100. In right-hand-side code, find all NavPath annotations within the resulting text region and mark each one as an ACE_NavigationPath annotation. Include in each of those annotations a feature ProjectName that contains the string of the ACE_Project annotation. Also, include a feature, “used”, set to the string “no” in each annotation as well. For those NavPath annotations that are subsequently associated with information of interest (e.g., dredging operations), this will be changed to “yes”. As part of the final clean-up, all those un-used NavPath annotations can be removed.

6.2.2.1 Navigation Path Characteristics – PRELIMINARY

NOTE: These rules are currently preliminary in nature; they have been developed and subjected to initial testing, but have not been thoroughly vetted. Data produced by these rules will not be delivered at the current time.

There are multiple characteristics that specify and describe navigation paths. The following rule sets identify and then associate these characteristics with the ACE_NavigationPath annotation.

1. The rules in ACE_characteristics.jape capture a range of numeric characteristics, such as length, width, and depth.

- a. Input: Token Lookup Split Number
- a. Logic: Logic: Look for patterns of words and numbers that identify various characteristics, including:
 - i. Depth: “depth of 8 feet”, marked as Depth
 - ii. Width: “8 feet wide”, marked as Width
 - iii. Length: “length of 1 mile”, marked as Length
 - iv. Area: “24 acres”, marked as Area

The rules also look for several characteristics associated with the removal and/or placement operations, rather than with the navigation path. The use of these annotations is described later in the document.

- v. Linear Measure: “from 1,800 linear feet of channel”, marked as LinearMeasure
- vi. Volume: “788,211 cubic yards”, marked as Volume
- vii. Weight: “7 234 short tons”, marked as Weight

Each of the characteristic annotations is created with three features

- viii. Value: numeric value of the quantity
- ix. Unit: the string of the units for the value
- x. used: set initially to “no” to indicate that the annotation has not been associated with a navigation path. This will be updated in subsequent rules.

- 2. The rules in ACE_navpath_chars.jape associate the identified characteristics with the corresponding navigation paths.
 - a. Input: ACE_NavigationPath Area Depth Length Volume Width ProjectChar Split
 - b. Logic: Look for an ACE_NavigationPath annotation followed by one to ten characteristics (Area, Depth, Length, Volume, Width). Including the Split annotation in the input set ensures that they are in the same sentence. Add features to the ACE_NavigationPath annotation for each characteristic that has not already been used (“used” feature set to “yes”). The feature names capture the type and the value/unit (e.g., WidthValue1, WidthUnit1). The digit at the end is used to allow for multiple characteristics of the same type (e.g., “the channel is 10-feet wide at the head and 12-feet wide at the mouth”) by tracking the number of characteristics of each type and appending a count to the feature names. Set the value of the “used” feature for each characteristic to “yes” so that it is not associated with any other navigation path.

6.2.3 Placement Structures

Within each site, there are references to structures where dredged material might be placed, including breakwaters, levees, and embankments. The initial Placement Structure processing looks to identify those elements and to associate them with the corresponding project. Later processing will remove references to those structures for which no placement information is provided.

The processing is accomplished in multiple phases.

1. The terms in ACE_placement_structure--gaz.lst identify base words for placement structures, including dikes, jetties, and walls. The gazetteer processing is configured to annotate the matching words as PlaceStruct.
2. The rules in ACE_placestruct_project.jape associate detected placement structures with the project information that precedes them.
 - a. Input: PlaceStruct ACE_Project ACE_District
 - b. Logic: Find ACE_Project annotation and subsequent PlaceStruct annotations, looking for an arbitrary number of the latter, currently set to a maximum of 200. In right-hand-side code, find all PlaceStruct annotations within the resulting text region and mark each one as an ACE_PlacementStructure annotation. Include in each of those annotations a feature ProjectName that contains the string of the ACE_Project annotation. Also, include a feature, “used”, set to the string “no” in each annotation as well. For those PlaceStruct annotations that are subsequently associated with information of interest (e.g., placement operations), this will be changed to “yes”. As part of the final clean-up, all those un-used PlaceStruct annotations can be removed.

6.2.4 Volume or Weight Removed and/or Placed

A primary target of the information extraction operations is the amount of material removed and/or placed during dredging operations. These amounts of material are initially marked with the Volume annotations or in some cases Weight annotations. Subsequent steps determine the type of operation associated with the amount of material and then associate it with a navigation path or placement structure.

The processing is accomplished in multiple phases.

1. The rules in ACE_volume_remove.jape identify cases where a volume represents something other than material that has been removed. For example, the text might describe material that needs to be removed in the future, such as “further excavation of about 3,700,000 cubic yards yet being needed”
 - a. Input: Token Volume Split
 - b. Logic: Look for specific phrases preceding or following the Volume annotation that suggest that it is hypothetical or future reference. In such cases, remove the Volume annotation.

2. The rules in ACE_weight_remove.jape identify cases where a weight represents something other than material that has been removed. For example, the text might describe the total material to be placed over several years, rather than an amount during the current fiscal year, such as “making a total of 9,365 tons placed under the contract”
 - a. Input: Token Weight Split
 - b. Logic: Look for specific phrases preceding or following the Weight annotation that it is not a weight removed in the current fiscal year. In such cases, remove the Weight annotation.
3. The rules in ACE_volume_weight_material.jape associate a type of material with the Volume or Weight annotation if such information is provided.
 - a. Input: Token Volume Weight Placement
 - b. Logic: Look for Volume or Weight annotations followed by sentence structures that describe the material, such as “of ordinary sand”. Allow for optional intervening measurement type (e.g., “bin measurement”). Add the string that describes the material type to the Volume or Weight annotation as a feature Material. If present, add the text of the measurement type as a MeasurementType feature.
4. The rules in ACE_placement_removal_terms.jape identify words and phrases that describe placement, removal or combined operations, such as “deposited”, “excavated” and “removed and placed”, respectively.
 - a. Input: Token Split
 - b. Logic: Locate specific words or phrases that describe the placement or removal activities. Mark the placement indicators as Placement annotations, the removal indicators as Removal annotations and the combined indicators as RemovalPlacement annotations.
5. The rules in ACE_placement_location.jape associate a location with a placement operation, looking for references in the same sentence. (Separate these rules from those that look across sentences to give priority to the closer reference, rather than GATE’s emphasis on the longest match.)
 - a. Input: Placement RemovalPlacement ACE_PlacementStructure Token Split Divider
 - b. Logic: Identify Placement or RemovalPlacement operations in the same sentence as an ACE_PlacementStructure annotation or for certain sentence constructs that describe the location of a placement (e.g., “placed in authorized disposal areas”). Add the string of the location as a feature PlacementLocation to the Placement or RemovalPlacement annotation. Also, copy the ProjectName feature from the ACE_PlacementStructure annotation to the Placement or RemovalPlacement annotation.

6. The rules in ACE_placement_location2.jape associate a location with a placement operation, looking for references in the preceding sentence.
 - a. Input: ACE_Placement ACE_PlacementStructure Split
 - b. Logic: Identify Placement or RemovalPlacement operations in the same sentence as an ACE_PlacementStructure annotation. Add the string of the location as a feature PlacementLocation to the Placement or RemovalPlacement annotation. Also, copy the ProjectName feature from the ACE_PlacementStructure annotation to the Placement or RemovalPlacement annotation.
7. The rules in ACE_navpath_volume1.jape look for navigation path and removal and/or placement references within the same sentence and associate them together.
 - a. Input: ACE_NavigationPath ACE_Removal ACE_Placement Split
 - b. Logic: Look for ACE_NavigationPath and ACE_Removal or ACE_Placement annotations within the same sentence in either order. Presence of Split prevents matches in subsequent sentences. Rules allow for multiple ACE_Removal or ACE_Placement annotations. Copy the ProjectName feature from the ACE_NavigationPath annotation to each of the ACE_Removal and/or ACE_Placement annotations. Change the “used” feature on the ACE_NavigationPath annotation to “yes”.
8. The rules in ACE_navpath_volume2.jape look for navigation path and removal and/or placement references in separate sentences and associate them together.
 - a. Input: ACE_NavigationPath ACE_Removal ACE_Placement Split
 - b. Logic: Look for ACE_NavigationPath in one sentence followed by one to four ACE_Removal or ACE_Placement annotations in the next sentence.
9. The rules in ACE_structure_volume.jape associates Volume removed or placed with the corresponding placement structure annotation when mentioned in separate sentences. This is only for cases where placement or removal has no associated NavPath from the previous phase.
 - a. Input: ACE_PlacementStructure ACE_Removal ACE_Placement Split
 - b. Logic: Look for ACE_PlacementStructure annotations followed by one or more (up to four) ACE_Removal or ACE_Placement annotations in the next sentence. Added a feature, AssociatedStructure, to each ACE_Removal and/or ACE_Placement annotations with the text of the ACE_PlacementStructure annotation. If present, copy the ProjectName and ProjectID information from the ACE_PlacementStructure annotation to the ACE_Removal and/or ACE_Placement annotations.
10. The rules in ACE_placement_removal.jape associates Volume with placement and/or removal operations.
 - a. Input: Placement Removal Volume Weight RemovalPlacement Split Divider

- b. Logic: Locate sentences that contain one or more Volume or Weight annotations along with a Placement, Removal or RemovalPlacement annotation. Annotate the Weight or Volume as an ACE_Placement, ACE_Removal or both, depending on which of the corresponding annotations was found in the sentence. Copy the features from Volume annotation, including Weight/Volume, Unit, Material, and MeasurementType. Annotate the Placement, Removal or RemovalPlacement as an ACE_Operation annotation.

6.2.5 Operations Characteristics

There are multiple characteristics associated with dredging and placement operations that need to be related.

6.2.5.1 Dates

The aim of the Date processing is to identify dates associated with dredging and placement operations. The information is often presented as a period, with a starting and ending date. This processing takes advantage of existing GATE processing that identifies dates and date ranges and annotates them as Date and DateRange, respectively

The processing is accomplished in multiple phases.

1. The rules in ACE_key_dates.jape identify dates in specific contexts that are likely to be associated with dredging operations, including starting date and period (of activity)
 - a. Input: Token Date DateRange Split
 - b. Logic: Look for patterns of words and dates indicate a specific context, such as “completed in” followed by a Date annotation or “period from” followed by a DateRange annotation.
 - i. Mark any DateRange as a Period annotation. Add features for the StartDate and EndDate. If the starting date string omits the year, as in “March 12 to April 29, 1953”, append the year portion of the ending date to the starting date.
 - ii. Mark individual dates as CompletionDate annotations.
2. The rules in ACE_operation_date.jape associate key dates with ACE_Removal annotation when the two are in the same sentence.
 - a. Input: ACE_Operation Period Split
 - b. Logic: Look for ACE_Operation and one or two Period annotations in the same sentence. Remark the Period annotations as ACE_Period. Copy the StartDate and EndDate features from the Period annotation to the ACE_Period annotation. Add a feature Operation ID that identifies the ACE_

Operation annotation by its annotation ID. Also, add features StartDate1, EndDate1, and if applicable StartDate2 and EndDate2, to the ACE_Operation annotation.

3. The rules in ACE_operation_date2.jape associates operations with corresponding dates in when the dates are in the previous sentence.
 - a. Input: ACE_Operation Period Split
 - b. Logic: Look for a Period annotation in one sentence followed by an ACE_Operation annotation in the following sentence. Remark the Period annotations as ACE_Period. Copy the StartDate and EndDate features from the Period annotation to the ACE_Period annotation. Add a feature Operation ID that identifies the ACE_Operation annotation by its annotation ID. Also, add features StartDate1, EndDate1 to the ACE_Operation annotation.

6.2.5.2 Fiscal Year

The FiscalYearOperation associates the document date with each of the operations that have been identified.

1. The rules in ACE_fy_operations.jape associate the document date with ACE_Operation, ACE_Placement and ACE_Removal annotations in the document.
 - a. Input: ACE_FiscalYear ACE_Removal ACE_Operation ACE_Placement
 - b. Logic: Look for ACE_FiscalYear annotation followed by one or more (currently set to a max of 1000) ACE_Removal ACE_Operation or ACE_Placement annotations. Add the text of the ACE_FiscalYear annotation as a feature called FiscalYear.

6.2.5.3 Costs

The aim of the cost processing is to identify costs associated with dredging and placement operations and associate those costs with the corresponding operations.

The processing is accomplished in several phases.

1. The rules in ACE_money_remove.jape removes any Money references that are of the form “XX cents”.
 - a. Input: Money
 - b. Logic: Identify Money annotations that contain the string “cents” and remove the annotations.
2. The rules in ACE_cost.jape identify monetary values in constructs that indicate they represent costs associated with dredging and/or placement operations.

- a. Input: Token Money Split
 - b. Logic: Identify Money annotations in phrases that indicate that the values represent operations costs, such as “at a cost and expenditure of \$41,309.64.” Mark these monetary values as DredgingCost annotations.
3. The rules in ACE_cost_sum.jape sum cost multiple consecutive values that are parts of the same project. For example, text of the form “at a cost of \$35,089 and \$39,078 respectively” would be marked as a single DredgingCost annotation with a TotalCost feature equal to the sum of the two original values.
 - a. Input: Token DredgingCost Money Split
 - b. Logic: Find DredgingCost annotations followed by the word “and” followed by a Money annotation. Extract the numeric portion of the DredgingCost and Money annotation and calculate the sum. Create a new DredgingCost annotation that spans the full extent and add a TotalCost feature set equal to the sum. Also, include a CostCalculated feature on the new DredgingCost annotation set to “yes”.
 4. The rules in ACE_form_feed_trim.jape identify DredgingCost annotations that span a FormFeed annotation. To avoid having formfeed characters with the cost annotation, move the annotation forward.
 - a. Input: DredgingCost FormFeed
 - b. Logic: Find DredgingCost annotations that contain a FormFeed annotation. Create a new DredgingCost annotation that terminates before the FormFeed and remove the old one.
 5. The rules in ACE_operation_cost.jape identify costs related to placement and/or removal operations.
 - a. Input: ACE_Operation DredgingCost Split
 - b. Logic: Identify DredgingCost annotations in the same sentence as or the sentence before an ACE_Operation annotation. Add the string of associated with the DredgingCost annotation to the ACE_Operation annotation as a feature called “Cost”.

6.2.5.4 Linear Measure

The description of some dredging operations contains information about the length of a navigation path that was dredged. This processing identifies and associates those values.

The processing makes use of the LinearMeasure annotations that were identified in ACE_characteristics.jape.

1. The rules in ACE_removal_linear_measure.jape associate LinearMeasure annotations with the corresponding ACE_Removal annotations.
 - a. Input: ACE_Removal ACE_Placement LinearMeasure Split

- b. Logic: Look for ACE_Removal annotations followed by a LinearMeasure annotation in the same sentence with no intervening ACE_Removal or ACE_Placement annotation. Copy the Value and Unit features from the LinearMeasure annotation to the ACE_Removal annotation as LinearMeasure and LinearMeasureUnit features respectively.
2. The rules in ACE_removal_area_measure.jape associate Area annotations with the corresponding ACE_Removal annotations.
 - a. Input: ACE_Removal ACE_Placement Area Split
 - b. Logic: Look for ACE_Removal annotations followed by an Area annotation in the same sentence with no intervening ACE_Removal or ACE_Placement annotation. Copy the Value and Unit features from the Area annotation to the ACE_Removal annotation as AreaMeasure and AreaMeasureUnit features respectively.

6.2.5.5 Station Range

The description of some dredging identifies the location of the dredging operations using station ranges. This processing identifies and associates those values.

The processing makes use of the LinearMeasure annotations that were identified in ACE_characteristics.jape.

1. The rules in ACE_station_range.jape identify references to station ranges along navigation paths
 - a. Input: Token Number
 - b. Logic: Look for constructs of the form “Station ## to Station #.#” where the pound signs are numbers, considering various cases and abbreviations. Create a StationRange annotation that covers the entire text, and add the first and second numbers (with accompanying plus sign in each case, if present) as Start and Stop features, respectively.
2. The rules in ACE_removal_station_range.jape associate StationRange annotations with the corresponding ACE_Removal annotations.
 - a. Input: ACE_Removal StationRange Split
 - b. Logic: Look for ACE_Removal annotations followed by a StationRange annotation in the same sentence or a StationRange annotation in one sentence with the ACE_Removal annotation in the next. Copy the Start and Stop features from the StationRange annotation to the ACE_Removal annotation as StationStart and StationStop features respectively.

6.2.5.6 Work Type

The description of some dredging and placement operations specify whether the operation is new work, maintenance or an environmental effort. This processing identifies and associates those values.

The processing occurs in two phases.

1. The rules in ACE_work_type.jape mark phrases that describe the three work types that have been specified to date.
 - a. Input: Token
 - b. Logic: Look for defined phrases that represent new work, maintenance or environmental effort. Mark any of these as WorkType annotation, using the rule feature to specify New Work, Maintenance or Environmental.
2. The rules in ACE_operation_work_type.jape associate the work type with the removal operation. (Do we need to extend to Placement???)
 - a. Input: ACE_Operation WorkTypeParagraph WorkType Split
 - b. Logic: Look for ACE_Operation annotations in the same sentence as a WorkType annotation in either order or ACE_Operation annotations within a WorkTypeParagraph annotation. When found, copy the “kind” feature value from the WorkType or WorkTypeParagraph annotation to the ACE_Removal annotation as a WorkType feature.

6.2.5.7 Paragraph Association

Certain operations that are described in the documents are either future or past work. One way to differentiate these is to look at the document structure, where current operations are described in a paragraph entitled “Operations during the current fiscal year” or something similar, while future operations may be found in a paragraph labeled “”. This processing tags placement and removal operations found in such paragraphs with text that identifies the type of paragraph in which it was found.

The processing occurs in multiple phases.

1. The rules in ACE_operations_paragraph_headers.jape mark phrases that match the beginning of such paragraphs/sections.
 - a. Input: Token Split LineFeed
 - b. Logic: Look for LineFeed followed by various specific phrases that mark the start of paragraphs of interest. Include variants of these phrases that occur over the years and some that are caused by OCR errors.
 - i. Operations and results during the fiscal year (During Fiscal Year)
 - ii. Conditions at end of fiscal year (End of Fiscal Year)
 - iii. Local cooperation (Local Cooperation)

- iv. Proposed operations (Proposed Operations)
- v. Existing project (Existing Project)
- vi. Others, such as “Terminal facilities), “Operating and care of...”, and “Effect of improvement”.

Annotate phrases other than “Others” as ParagraphStart with a feature called type that contains the text shown in parentheses above. Annotate phrases that match the other category as ParagraphEnd.

2. The rules in ACE_operations_paragraph.jape mark the paragraphs/sections that generally follow immediately after the current year operations.
 - a. Input: ParagraphStart LineFeed ParagraphEnd ACE_District ACE_Project
 - b. Logic: Look for ParagraphStart followed by up to 25 lines. (Rule will stop capture at another instance of ParagraphStart or an instance of ParagraphEnd, ACE_District or ACE_Project. Annotate the entire extent as ACE_Paragraph, copying the type feature from the ParagraphStart annotation.
3. The rules in ACE_operations_paragraph2.jape extend the length of paragraphs/sections that were marked by the previous rule ACE_Paragraph annotation is followed by a ParagraphStart annotation within 25 lines.
 - a. Input: ACE_Paragraph ParagraphStart LineFeed ParagraphEnd ACE_District ACE_Project
 - b. Logic: Look for ACE_Paragraph followed by up to 25 line and then followed by a ParagraphStart annotation. (Rule will not fire if there is an intervening instance of ParagraphEnd, ACE_District or ACE_Project. Remove the old ACE_Paragraph annotation and annotate the new extent as ACE_Paragraph, copying the type feature from the ACE_Paragraph annotation.
4. The rules in ACE_operations_in_paragraph.jape look for operations annotations that are found within an ACE_Paragraph annotation and copy over the information from the type feature.
 - a. Input: ACE_Paragraph ACE_Placement ACE_Removal
 - b. Logic: Look for ACE_Paragraph annotations that contain either at least one ACE_Removal or ACE_Placement annotation. Extract all ACE_Removal and ACE_Placement annotations found in the ACE_Paragraph annotation. Copy the text of the type feature from the ACE_Paragraph annotation to each of the ACE_Removal and ACE_Placement annotation as a Paragraph feature.

6.2.6 Multi-part Operations

The description of some removal operations includes a breakdown into components. An example of this is:

A total of 2,333,419 cubic yards of material was removed, of which 1,838,850 cubic yards was new material and 494,569 cubic yards was for maintenance.

This processing, in a single phase, identifies such breakout situations and marks the main and constituent components.

1. The rules in ACE_removal_constituents.jape identify phrases that multiple removal operations with a primary-subsidary relationship
 - a. Input: ACE_Removal Token
 - b. Logic: Look for sentences that contain multiple ACE_Removal annotations with intervening words that indicate the constituent relationship. (So far, this is limited to “of which”???) Update the subsidiary ACE_Removal annotations with two features: Constituent, set to “yes” and MainID, with the annotation ID of the primary ACE_Removal annotation.

6.2.7 Districts

The Chief’s report documents follow the hierarchy of Districts -> Projects -> Operations, where the District information is frequently provided in un-numbered section headings. This processing identifies the districts.

The operation is spread across the following phases.

1. Gazetteer file ACE_district—gaz.lst contains a list of the current Army Corp districts, found on the Army Corp Web site. This has been augmented with select prior districts that have been identified while processing documents. In each case, the information is augmented with information on the stated district and the corresponding present-day district as follows:

ALASKA:StatedDistrict=Alaska District:PresentDayDistrict=Alaska District

The gazetteer processing is configured to mark each item in this file as a District annotation, with StatedDistrict and PresentDayDistrict as features populated appropriately.

2. The rules in ACE_district_headings.jape identify structures in the document that look like they might be district headings.
 - a. Input: Token Number LineFeed Split ACE_Project
 - b. Logic: First, avoid tagging any text that is already marked as an ACE_Project annotation. Also, avoid tagging text that appears to be an Appendix or text that names a specific waterway. Then, look for the full line of text that is all-caps or upper-initial, with the optional leading number and trailing parenthetical statement. Mark the output as an ACE_District annotation with the text of the district, normalized to remove multiple spaces, carriage returns and line feeds, as a DistrictName feature and a feature “used” set to “no”.

3. The rules in ACE_district_name.jape weeds out ACE_District annotation that are not of the correct form
 - a. Input: ACE_District District
 - b. Logic: Find ACE_District annotations that contain a District annotation. Copy the StatedDistrict and PresentDayDistrict features from the District Annotation to the ACE_District annotation. Add a feature “real” to the ACE_District annotation with a value of “true”. Find ACE_District annotations that contain the strings DISTRICT and ADDITIONAL and do nothing. Finally, find an ACE_District annotations that contain the string “DISTRICT” or “DIVISION” and add a feature “Real” to the ACE_District annotation with a value of “true”.
4. The rules in ACE_district_group.jape remove ACE_District annotations that contain a group designation.
 - a. Input: ACE_District GroupString Sentence Split
 - b. Logic: Look for ACE_District annotations that contain or are followed by a Group annotation. (Group annotation, identified in ACE_group.jape, marks text of the form “(GROUP %)” where % is a capital letter as GroupString annotation.) Remove the ACE_District annotations in such cases.
5. The rules in ACE_district_remove.jape identify those ACE_District annotations that have not been identified as real and removes them.
 - a. Input: ACE_District Token ACE_Project
 - b. Logic: Ignore ACE_District annotations that have a feature “Real” set to “true”. For all other ACE_District annotations, remove the annotation.
6. The rules in ACE_district_refine.jape removes ACE_District and ACE_Commission annotations that are not the first in a sequence.
 - a. Input: ACE_District Token ACE_Project
 - b. Logic: Look for ACE_District or ACE_Commission annotations identified by the rules RightDistrictHeader_First or RightCommissionHeader_First respectively, followed by up to fifty ACE_District or ACE_Commission annotations that are identified by some other rule. Remove the ACE_District or ACE_Commission annotations on all but the first.
7. The rules in ACE_district_operation.jape, ACE_district_placement.jape and ACE_district_removal.jape associate a district with each ACE_Operation, ACE_Placement and ACE_Removal annotation. These rules treat ACE_Commission as an alternate for ACE_District

Note that the, due to the large number of entities, the processing for ACE_Operation and ACE_Removal are designed to be memory efficient. In particular, they avoid doing the matching on the LHS of the rule, but rather perform the matching in the RHS Java code.???

- a. Input: ACE_Removal ACE_Placement ACE_District ACE_Commission
 - b. Logic:
 - i. Look for an ACE_District annotation followed by one or more ACE_Removal and/or ACE_Placement annotations (currently set to 300 maximum). Copy the DistrictName, PresentDayDistrict and StateDistrict features from the ACE_District annotation to each of the ACE_Placement and ACE_Removal annotations. Add a feature “used” to the ACE_District annotation set to “true”.
 - ii. Look for an ACE_Commission annotation followed by one or more ACE_Removal and/or ACE_Placement annotations (currently set to 300 maximum). Copy the CommissionName feature from the ACE_Commission annotation to each of the ACE_Placement and ACE_Removal annotations. Add a feature “used” to the ACE_Commission annotation set to “true”.
8. The rules in ACE_district_remove_unused.jape remove ACE_District annotations that are not associated with a removal or placement operation
- a. Due to change in requirements, this is commented out for now.

6.3 Storing Results

The standard GATE output includes the entire input document and all contained annotations, which is not needed for this application. Instead, custom code was written as part of the Named Entity processing to write the features and values from specific annotation types to files.

1. The rules in ACE_write_results.jape parse the annotation set, find annotations of specific types and write them, with their features, to particular files.

During the write operations, the various IDs are prepended with a hash code of the source file URL. This is done in order to ensure that the IDs are globally unique and therefore the results of processing of multiple files can be merged into a single file without reference conflicts.

- a. Input: ACE_Placement ACE_Removal
- b. Logic: The match here simply looks for an ACE_Placement or ACE_Removal annotation. As long as there is at least one of those, then the results are interesting. The Java code then gets all the annotations of particular types from the entire file and writes them out as follows:
 - i. ACE_Removal: Construct records made up of the following features from the ACE_Removal annotation: Type, FiscalYear, CurrentFiscalYear, StatedDistrict, PresentDayDistrict, ProjectName, RemovedFromNavPath, VolumeRemoved, VolumeRemovedUnit, MeasurementType, MaterialRemoved, LinearMeasure,

LinearMeasureUnit, StationStart, StationStop, WorkType, OperationID, DistrictID, ProjectID, Offset. Type is set to “Removal”, Offset is the starting position of the annotation from the beginning of the document and the remaining items are ACE_Re moval features. One record of this type is written to the output file for each ACE_Re moval annotation in the document. The files are named:

Removal_{FiscalYear}_Volume{DocVolume}_Part{DocPart}

Where FiscalYear is the string value of the ACE_FiscalYear annotation and DocVolume is the string value of the ACE_DocVolume annotation. If any of these values are missing, the header and value are omitted from the file name. If all of them are missing, a randomly generated integer is added to the end of the file name after the “Removal_” prefix.

6.4 Galveston Processing

For the period from 1968 until 2000, the dredging information for the Galveston district was not described in the prose of the document, but rather included in tabular form at the end of the section. The information extraction process was modified to specifically capture information from these tables. Note that the text files that included this tabular data were manually edited to ensure that the data was in the proper format and order.

The Galveston operations involved the following eight steps.

1. Project Marking – a list of projects in the Galveston district was generated. The Gazetteer process was used to tag these projects as GalProject
2. Dredge References: The rules in ACE_gal_dredge_ref.jape tag references to dredging vessels as generally found in the Galveston tables, such as (U.S. hopper dredge A. Mackenzie). These instances are annotated as GalDredge.
3. Navigation Paths: the rules in ACE_galv_nav_path.jape detect references to navigation paths in the Galveston tables. These are found either following a GalProject reference or following two numbers and preceding a date range. The navigation paths are annotated as GalNavPath.
4. Galveston Data Row: the rules in ACE_galveston_row1.jape identify rows in the Galveston tables. They start by verifying that the fiscal year of the document is within the range of 1968 to 2000. If the document is from outside of that range the process stops. It then detects a pattern of GalNavPath, followed by optional GalDredge, followed by one, two or three date ranges followed by two numbers. ACE_Operation and ACE_Re moval annotations are generated, with information from the date ranges used as start/end date features, the first number is added as the volume and the second number as the cost. The GalNavPath is re-annotated as an ACE_NavigationPath.
5. Incorrect Projects: In certain cases, the Galveston NavPath is tagged as a Project instead. The rules in ACE_galveston_fix_project.jape correct for this. If the ACE_Navigation_Path contains a GalProject, the GalProject annotation is removed.

6. ACE_galv_project: Finds the references to projects in Galveston Tables and annotates them as ACE_Project.
7. ACE_galv_project_navpath: Removes navigation path annotations from within ACE_Project annotations in Galveston tables.
8. ACE_galv_navpath_project: Associates detected navigation paths with the corresponding projects within Galveston tables.