MDPrep: Data Preparation for Metadata Management

Hiba Khalid
Université Libre De Bruxelles
Brussels, Belgium
Hiba.Khalid@ulb.ac.be

Esteban Zimányi
Université Libre De Bruxelles
Brussels, Belgium
Esteban.Zimanyi@ulb.ac.be

ABSTRACT

With the increased amounts of data production, a significant surge has appeared in metadata generation as a part of the process. This metadata can provide meaningful insights, leading to improved data analytics, data integration, and resource management. However, due to many variations, users and tools do not follow the standards during recording metadata, resulting in various inconsistencies, such as missing attribute information, missing publishing URL, lack of provenance, etc. In addition, even recorded metadata may contain inconsistencies, such as multiple value formats, values with special characters, incorrectly entered values, etc. Preparing metadata can improve the user experience during data management tasks by addressing the above mentioned inconsistencies.

This paper discusses the usability and applicability of data preparation techniques for enhancing metadata. We approach the problem by (1) detecting and identifying metadata elements and structural metadata issues, (2) applying a keyword-based approach for preparing metadata elements and syntax-based approach for preparing structural metadata issues, (3) comparing the outcome for improved readability and reusability of prepared metadata files.

KEYWORDS

data preparation, metadata management, metadata representation, metadata categorization

1 INTRODUCTION

Syntactically and semantically enhanced metadata can help users and applications with various tasks, such as searching for datasets in web repositories [2, 8], schema matching [6, 22], integrating data [13, 17], exploring data lakes [23, 25], etc. Online repositories contain metadata in various formats published by government data publishers, commercial data providers, and scientific projects. Since metadata in online repositories is automatically generated based on publisher-provided details, it contains many quality issues, such as missing titles, missing attribute names, dates with different formats, etc. These quality issues occur differently in various file formats, e.g., incorrect hierarchy in JSON file format, data arrangement to facilitate human visual inspection in TXT files, syntactically incorrect attribute values in CSV files due to its loose standard, etc. It primarily depends upon the organization or data publisher that records, manipulates, and distributes data. In many cases, standards [7, 14] designed for collecting and recording metadata are not followed. Therefore, depending on the task, a metadata file may need to be restructured and prepared before it is made available for custom operations, resulting in a manual user effort for each file. By applying data preparation techniques, we can address the aforementioned issues, improve file reusability, and increase the quality of metadata files.

1.1 Data Preparation

Data preparation is the pre-processing operations performed in the early stages of a data processing pipeline [9, 12, 26]. Applying data preparation techniques can provide various advantages, such as detecting errors, uniform formatting of values, normalizing numeric values, etc. Typical data preparation tasks involve (1) data cleaning, (2) data transformation, (3) data standardization, and (4) data enrichment.

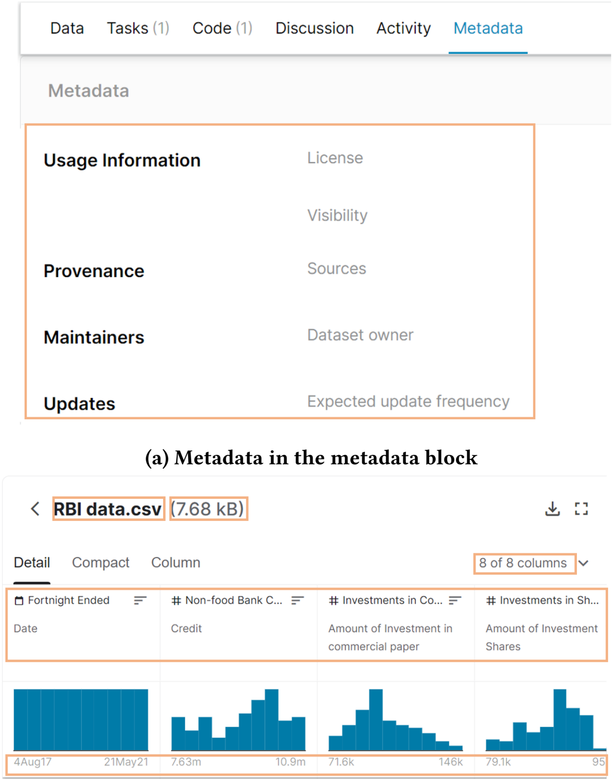
In the context of our research, we define and leverage aspects of data preparation to better fit the needs and challenges encountered in metadata files. We use a list of preparation tasks defined as data preparators [9] from the literature and apply them to prepare metadata for metadata management. A fundamental aspect of our research is to repeat the experimental results of different approaches (data transformation, value rectification, format standardization, etc.) and compare the quality of the metadata file before and after their application. For example, a value transformation is a well-known problem in the literature [1, 11], and here we enable repeatability by applying it to different usecase. To observe the applicability of data preparators [9], we collected metadata files from three open data repositories: Kaggle[1] , UKGov[2], and DataGov[3]. These three resources contain various metadata files in different formats and different types of metadata issues (see Section 2 for details) that can cause problems in data management tasks. For instance, in resource management, inconsistent metadata can create issues in legacy maintenance, ontology alignment, resource backups, duplicate data sources, and updates and changes to both resource and the metadata file. The issues prevalent in metadata files are discussed in detail in Section 2.

To further investigate metadata issues, we examined data that we crawled from Kaggle. Out of 563 randomly downloaded files, we manually checked and found that 512 files required some level of data preparation.

Figure 1 is a snapshot downloaded from Kaggle[4], showing an example of metadata arrangement where the metadata is spread across multiple data blocks. Figure 1a shows the metadata block on the data source that contains the relevant information about the dataset. However, this is not the only place that contains metadata. The data block also contains metadata in various sections, as shown in Figure 1b. For space reasons, we omit the details of the "column" section in Figure 1b, which also contains all the information about the attributes.

For instance, a data scientist has been handed a data integration task that requires this dataset shown in Figure 1. Since metadata is not consolidated and downloadable, a data scientist has to first scan, scrap, or manually consolidate the available metadata across different tabs and pages, i.e., exploration and identification of potential metadata. Once all the metadata is consolidated in

[1] https://www.kaggle.com/
[2] https://data.gov.uk/
[3] https://www.data.gov/
[4] https://www.kaggle.com/sandraphari/flow-of-financial-resources (February 2022)



(a) Metadata in the metadata block

(b) Metadata in the data block

Figure 1: Figure 1a shows that metadata has its own block where the user should expect to find all relevant metadata, while Figure 1b contains column-level and cell-level metadata, showing how metadata is distributed across multiple locations, resulting in prior manual work for subsequent extraction operation.

one file, an expert has to label different types of metadata with relevant headers, keywords, and relevant tags for automated tasks regarded as collection and labeling processes. Thus, a data scientist needs to perform the above tasks by manually consolidating a reusable metadata resource and then evaluating file eligibility for integration.

Several problems in metadata files are solvable by applying data preparation techniques and automating this process. For example, a metadata file structure can improve by (1) re-arranging items in a file, (2) removing empty blocks, (3) correcting date formats, (4) standardizing cell values, and (5) filling missing attribute information, etc. By resolving these errors, relevant problems can improve the existing information in metadata files, resulting in improved metadata for data-driven tasks, such as data discovery [2, 3], data lake resource management [20, 27], data augmentation and integration [17, 21], and data provenance and lineage [4, 28].

1.2 Metadata Preparation Categories

In this section, we categorize the applicability of data preparation to metadata into two main categories, (1) Syntactic data preparation (2) Semantic data preparation.

1.2.1 Syntactic data preparation . Syntactic data preparation is relevant for correcting erroneous information, pre-processing, and removing redundant or ill-formed information. It deals with the file's structure and content, such as data row boundary detection, cell value boundary detection, and formatting. For example, converting a date from one format to another requires a syntactical understanding of the date components. A list of syntactic data preparators can be found in Table 3.

1.2.2 Semantic data preparation. Semantic data preparation consists of the data preparators identifying the data or file contents based on the keywords, their placement in the file and their domain relevance. It not only requires understanding of the structure, but also the semantics based on the underlying operation. For example, standardizing the value to a common cluster requires prior semantic understanding, e.g., replacing *new york, nyc, NY* to *NEW YORK CITY*. Table 3 lists a set of semantic data preparators.

The aforementioned steps present a unique opportunity in data preparation applications for metadata resources. With the help of various data preparators, it is possible to transform the misplaced and raw metadata into clean and reusable resources. In a nutshell, we prepare data for metadata management by systematically applying existing data preparators and observing the quality of metadata before and after preparation,

Our paper makes the following contributions:

- A resource survey to describe open data portal metadata and its quality issues.
- A process, to apply data preparators for metadata preparation.
- A system, MDPrep, to automatically transform raw metadata to prepared metadata.
- A set of experiments to validate MDPrep results and its applicability.

The rest of the paper is organized as follows: Section 2 describes the process of metadata collection and manual inspection of errors in collected files. Section 3 illustrates the workflow of the MDPrep and explains its main modules. Section 4 presents the experimental evaluation of MDPrep. Section 5 discusses related work in the field of data preparation for diverse domains, and Section 6 concludes our study.

2 RESOURCE SURVEY

This section describes the methodologies adapted to collect and consolidate metadata files from open data repositories. This section also walks through the description and nature of collected metadata files.

2.1 Metadata Collection and Consolidation

During our manual survey, we aimed to accomplish the following:

- Metadata Sniffer: to discover and identify how dispersed the metadata is in a given resource.
- Metadata Keyword Management: to extract keywords from descriptions, tags, and metadata files to create a category of keywords and populate a knowledge-base.
- Metadata Preprocessing: to detect structural and semantic inconsistencies in metadata files that may hinder file processing.

We exported metadata files from three open repositories, Kaggle, UKGov, and DataGov. We exported the available metadata files from each repository and manually collected and consolidated dispersed and embedded metadata. We created a metadata resource and named it MetaVader. MetaVader contains a total of

1123 metadata files from the three mentioned repositories and is a mixed collection of files representing different datasets (see Table 4).

2.2 Resource Inspection

During the collection and consolidation process, we manually evaluated each file and labeled the structural and semantic issues present in the files. This manual process involves manually inspecting each metadata file for labeling and identifying relevant elements from the set. As a part of this process, we manually deduced the structural issues prevalent in each metadata file from each resource. This includes detecting layout issues, empty paragraphs, foreign languages in a metadata file, embedded data within the metadata file, incorrect spacing, incorrect formatting, etc. The broader inspection categories included the following:

- Identifying file structural issues
- Identifying missing metadata
- Identifying missing meta-metadata, or incorrectly labelled meta-metadata
- Identifying underlying semantic issues

2.3 Resource Statistics and Pre-processing

As mentioned earlier, the metadata we collected was spread across multiple sites for the same resource. To collect and use this metadata, we performed a "layout data preparation". This involves identifying data order, information order, and content sequencing. In our context, we performed layout data preparation as a manual step for metadata collection, considering the following aspects:

- Manually detect and extract metadata from the metadata section in the header. Since there is no consolidated downloadable metadata file, it must be extracted manually.
- Extract metadata header and its corresponding value such as 'Provenance' is the metadata header. Its value is the web link to its resource tagged with meta-meta data 'sources.' The available information depends on the file type under consideration.
- Maintain and add extracted metadata to a file or resource for further processing and reusability.
- Explore other tabs and pages to identify potential metadata for extraction and metadata maintenance.
- Manually extract keywords from dataset description.
- Extract headlines, keywords, acknowledgments, attribute lists, attribute types, attribute descriptions, and other available metadata from the data tab.

In addition, each data source in MetaVader had its challenges. To address the cleaning issues in the metadata files, we evaluated each file from each resource (Kaggle, UKGov, DataGov) to collect statistics on the nature and type of inconsistencies. We accessed each type of metadata file and captured the following characteristics:

- Resource Description: we manually added meta-information regarding each metadata file, data type, domain, and file size.
- Resource Statistics: we added context to information available in the files by identifying the type of inconsistencies present in each file. We categorized them by file type and further by inconsistency type.
- Differentiated Resources: we inspected and recorded how different resources were from each other and in what aspects, such as the type of data, level of inconsistencies, types of inconsistencies, amount of information, etc.
- Preparation Levels: we recorded the level and type of preparator needed for the corresponding inconsistency type for each file in MetaVader.
- Reader Facilitation: we incorporated both pros and cons about each file in MetaVader to build user understanding around available information, prevalent issues, effects of inconsistencies, and advantage points.

Finally, we collected some noticeable issues observed in low-quality metadata files and listed them in Table 1. The table shows meta and meta-meta level issues related to syntax and semantics across different file formats. We only focus on spreadsheets in this research.

2.4 Resource Syntactic and Semantic Preparation

Each file of a particular resource had both syntactic and semantic inconsistencies. Our goal was to establish a meaningful discourse about the information in a given metadata file. To achieve this, we manually reviewed each file and categorized the problems in the metadata based on syntactic and semantic preparation. As the name implies, syntactic preparation involves structural changes in data, i.e., data type conversions, trimming whitespace, value formatting, etc. Semantic preparation involved understanding the meaning of operations, i.e., understanding and converting date components, standardizing values to a common cluster, determining semantic roles for attributes, etc.

3 MDPREP METHODOLOGY

In this section, we build an understanding of each process involved in the workflow of MDPrep (see Figure 2). Given a raw, unprepared metadata file, MDPrep performs the following tasks.

- Metadata Content Tagging
- Metadata Preparation
  - Syntactic Metadata Preparation
  - Semantic Metadata Preparation

3.1 Metadata Content Tagging

This section describes the tagging of metadata content that the system later uses for metadata preparation. It highlights the issues and techniques developed to address metadata inconsistencies in the collected metadata files. MDPrep starts with identification and tagging the type and content available in the metadata file. In our use case, we define metadata content into six main categories. We define these elements to assist in detecting quality issues and the type of content available for utilization (see Table 2). We can identify the placement and usage of information in a particular metadata file by defining these elements. It provides an overview of how spread out and variable the metadata content is. The goal is to identify as many elements as possible in a metadata file. It increases file usability and readability. In Figure 3 we illustrate how useful element identification is when it comes to metadata content. The identified elements can be rearranged, re-ordered, annotated, or commented on for any task. Each element is a detection category for the type of metadata available in files. For example, 'E1' is an automated method that identifies meta

Content Type	Content Description	JSON	Spreadsheets	CSV	TSV	Embedded	Text	Other
Representation	affiliations, institutes, author names, email, phone, etc.	T	T	T	T	T	T	
Connections	hyperlinks, references, tags, keywords, etc.	T	T	T	T	T	T	T
Arrangement	volume no, pages, schema, file lists, attribute lists,ssn, etc.	T	T	T	T	T	T	T
License	publisher, data provider information, etc.		T	T	T		T	
Labels	missing data labels, missing headers, sub headers, titles, subtitles, etc.		T	T	T			
Use Me	missing user guides, missing comments, etc.	T	T	T	T		T	
Meta Metadata	column descriptions, attribute dependency, primary foreign key descriptions, etc.	T	T	T	T	T		T
Structural	data organization, attribute management, connected files, resources, etc.	T	T	T	T	T		
Semantic	naming conventions, column stats, attribute distribution, acronyms, etc.		T	T	T	T	T	

Table 1: Boolean representation of classifiable metadata content types with respective descriptions and prevalence in different file formats acquired for experimentation. We filled in the corresponding cell as true (T) if 90% of the collected files for the specific format contained the items listed in content description.

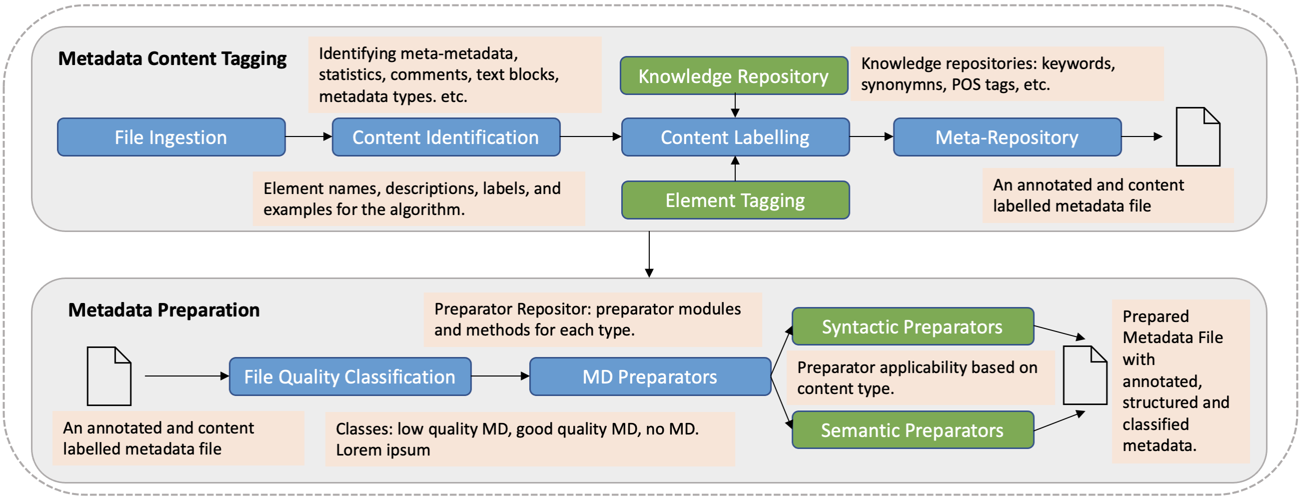

Metadata Content Tagging
Identifying meta-metadata, statistics, comments, text blocks, metadata types. etc.
Knowledge Repository
Knowledge repositories: keywords, synonymns, POS tags, etc.
File Ingestion
Content Identification
Content Labelling
Meta-Repository
Element names, descriptions, labels, and examples for the algorithm.
Element Tagging
An annotated and content labelled metadata file
Metadata Preparation
Preparator Repositor: preparator modules and methods for each type.
Syntactic Preparators
Prepared Metadata File with annotated, structured and classified metadata.
File Quality Classification
MD Preparators
Preparator applicability based on content type.
Semantic Preparators
An annotated and content labelled metadata file
Classes: low quality MD, good quality MD, no MD. Lorem ipsum

Figure 2: The workflow of MDPrep

metadata embedded in files, identifies meta-meta values, and consolidates them.

Similarly other element categories are responsible for detecting, annotating, and arranging metadata content. To recognize and label each element type, we use a meta-repository containing definitions, descriptions, and examples.

We use a keyword-based approach that recognizes and labels metadata elements for automatic identification. We first create a knowledge base that we use to train our algorithm. When creating our knowledge base, we first manually label the data. For this purpose, we had experts manually label our data. The next step is to use this labeled data and create negative examples to make the system error-resistant. This process serves as automated pre-processing for our syntactic and semantic preparator modules.

3.2 Metadata Preparation

First, this section describes the importance of data preparators for metadata. Then, it discusses how MDPrep leverages syntactic and semantic preparators for metadata preparation.

3.2.1 *Data Preparators for Metadata.*

As discussed, data preparation is the process of transforming and cleaning before a file or collection of files can be processed; we propose metadata preparation as a process to enhance, improve, and quantify the available metadata content. To define and understand metadata preparation, it is critical first to define the prospects that can be addressed by metadata preparation.

As mentioned before, a fundamental aspect of our research is to repeat the experimental results of different approaches (data transformation, value rectification, format standardization, etc.) and compare the quality of the metadata file before and after their application. To establish a ground base for metadata preparation, we leverage a set of data preparators from literature [9] that can target several prevalent metadata issues. For example, incorrect schema information can cause problems in many data-driven tasks, such as data integration. Similarly, an inconsistent date format can make it challenging to understand the meaning of date components when querying data. Leading and trailing spaces that are part of the numeric value can pose a problem for the

Metadata Element	Element Description	Element Example
E1	Meta metadata	metadata cell or a block of cells to explain metadata of a file
E2	Disparate metadata	metadata placed in a file referenced for another file
E3	Description	metadata element with concrete description
E4	Empty	metadata file cells or block of cells are empty
E5	Content	metadata content
E6	Missing	metadata file that has missing elements, e.g., publisher:__EMPTY__

Table 2: Metadata element set with element description and examples.

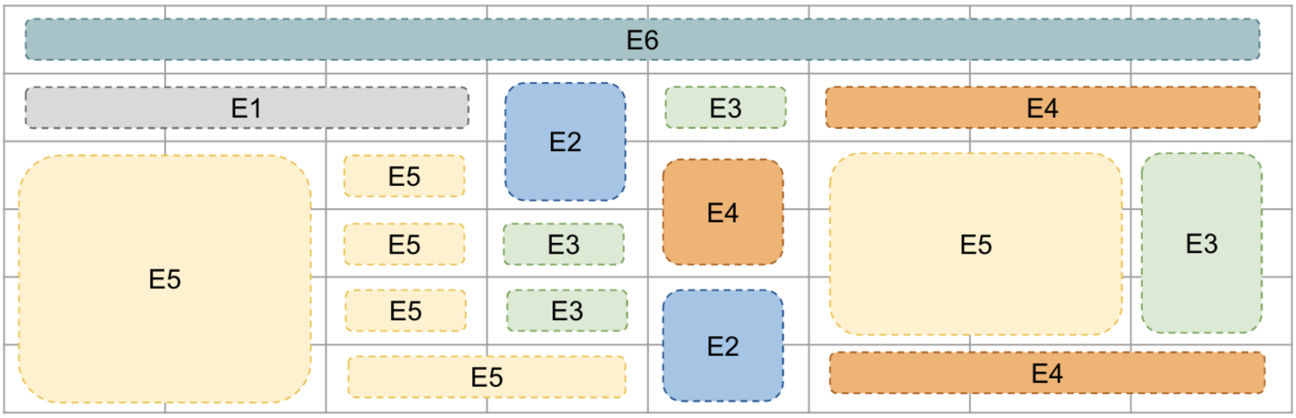


Figure 3: A synthetic overview of customized metadata element set based on the content of the metadata files from all sources. The Figure represents the metadata placement in files in MetaVader and shows the tagging of these elements during the process.

data analysis tool when retrieving this data and would store such values as text instead of a number. Preparing such issues before using metadata would support the data processing pipeline.

In the next sections, we discuss the capacity and role of our metadata preparators.

3.2.2 *Syntactic Preparators.*

Table 3 shows the list of syntactic preparators that our system supports. By our definition, a metadata preparator aims to target quality issues prevalent in metadata files. A preparator is a specific method to target content discrepancies to produce more meaningful and readable information in a metadata file. A collection of preparators target metadata discrepancies and enhance metadata files' overall readability and comprehensiveness. Each preparator defined serves a purpose in improving metadata file readability and reusability. For example, *value rectification* is an important preparator to ensure the same standards are followed throughout a metadata file, and it also sets an expectation of the format available in the data itself. One of the most commonly deployable preparators is *delete and keep empty*. This preparator scans a metadata file and identifies empty blocks, whitespace, and a heading with no text underneath. This is particularly helpful and useful as it reduces ambiguity and identifies metadata file headers or content that requires attention, such as the manual addition of comments, subscripts, and definitions.

While some preparators have predefined criteria for cleaning, such as trimming whitespace, others must be processed incrementally to select the final value, such as date transformation. We use the date transformation module from the literature [11] and apply programming transcripts available in online repositories. However, to limit the scope, we have defined a set of patterns for dates that we allow during our transformations.

3.2.3 *Semantic Preparators.*

Our research defines semantic preparators as methods that identify and understand the metadata values' content before applying the transformation. Table 3 lists the semantic preparators supported by MDPrep.

When using the semantic preparator, our system understands the context before cleaning up the format. Similar to the syntactic preparator, we also refer to the literature for the semantic preparator. For example, for *value standardization*, we first need to detect the duplicates. For duplicate detection, we use a clusters-based approach to identify entities representing the same real-world data [24].

Similarly, *extraction of parts* from cell values also requires semantic differentiation of the parts of the values before splitting it into different segments. For example, values containing a delimiter character not escaped can cause a shift problem, resulting in incorrect segment identification, such as cell value `1,000`; the correct representation of this cell value will be `"1,000"` or the delimiter character for the cell value separator should be other than ",", such as ";" to avoid column shift problem. To understand this, we leverage the row and type pattern approach from [29] and parse components correctly before applying the split function for segments.

3.3 Prepared Metadata File

The final phase in MDPrep is to create a reusable layout of prepared metadata files containing the appropriate metadata elements. The prepared metadata file has the following components:

Preparator	Preparator Description	Example
Special character removal (SCR)	Detects and removes special characters and diacritics that do not add value to available metadata	⟨Brussels⟩ ==>Brussels
Change letter case (CLC)	Allows for improved readability by identifying sentences, ends of sentence, and using capitalization techniques to better fit the metadata content in file	Vldb ==>VLDB
Acronymization (ACR)*	Facilitates assigning meaningful acronyms and by detecting existing acronyms	Special Interest Group on Management of Data ==>SIGMOD
Standardization (STDN)*	This improves content consistency within a metadata file	New york, NEW York, NYC ==> NEW YORK CITY
Normalize numeric values (NNV)	Transform numeric values to a common scale	55.551, 10.3120, 12.18 ==> 55.5, 10.3, 12.9
Trimming (TR)	Remove leading and trailing extra whitespace from cell values	␣12.31 ␣==>12.31
Value rectification (VR)	Identifies and changes data value based on column format	Dataset Publish Date: 11/11/2001 10 Feb; 2010 ==> Dataset Publish Date: 11/11/2002 10/02/2010
Sort data (SD)	Sort data in ascending, descending or alphabetical order	Updates: 1/02/2006, 6/9/2001 ==> Updates: 06/09/2001, 01/02/2006
Mismatched data (MSD)	Identifies values that are inconsistent with a header definitions and other data values	0.03, 0.05, text ==>text
Mistyped data (MD)*	Identifies and rectifies spelling issues and common typing mistakes	Chicgao ==>Chicago
Permitted characters (PC)	Searches and regulates allowable characters in files It is based on collection of special characters.	Prov.: $ https://XYZ.com/ ==> Prov.: https://XYZ.com/
Renaming (RE)*	Allows renaming columns; cell values; headers; comment section; etc.	Addr. ==>Address
Split (SP)	Splits columns based on pre-defined criteria	14482, Potsdam ==> 14482 Potsdam
Merge (MR)	Allows rows and column merges based on pre-defined criteria	Dataset Owner: Shivam Dataset Owner: Daniel ==> Dataset Owner: Shivam & Daniel
Delete or keep empty (DKE)	Allows the user to either delete or keep empty values; rows and empty header sections	1,Bob,␣,Germany ==> 1,Bob,Germany
Extract parts (EP)*	Allows extraction of textual metadata from spreadsheets, and allows extracting specific column(s)	10001, New York, USA ==> Zip Code: 10001, City: New York, Country: USA
Edit and replace data (ERD)*	Allows the user to either edit or completely replace metadata values with new information	Visibility: Public ==>Visibility: Private

Table 3: An example explanation of the proposed metadata preparators from literature [9] (syntactic and semantic*) to improve the understandability and reusability of metadata files.

- File Summary: this includes the set of preparators we applied during the preparation of a raw file. If the metadata file already had a summary (cleaned file), this is the first content we observe in the prepared metadata files. It also includes file specification metadata such as file type: XLSX, total records, and value types: text, number, mixed, etc.
- Content Summary: the outputs obtained from metadata preparators provide consolidated information set with different types of metadata and its values. The content summary includes keywords, tags, meta-meta data, and contact details (if available in the original file). It also includes a list of acronyms used in the file (added from semantic preparation), column summaries, data value ranges, etc.

- Metadata Content: this section in the file includes an arranged, prepared, and cleaned format of the original metadata file with relevant metadata tags, types, and annotation details.

4 EXPERIMENTS

In this section, we first describe the datasets we used for our experiments. We then explain the performance evaluation of MDPrep.

4.1 Datasets

The experiments were conducted on metadata collection to understand file structure, organization, content change, and preparation. In the context of this research paper, we limit our experimentation to spreadsheet datasets. It contains a collection from Kaggle, DataGov, and UKGov resources. Regarding files gathered from Kaggle, it is essential to note that the metadata files from Kaggle comprise Kaggle resource descriptions, downloadable files, and misplaced metadata. We experimented with a total of 1123 metadata files from the three aforementioned repositories, and they were added to a "mixed collection" for experiments of data preparation techniques. Throughout this paper, we will refer to this as the "MetaVader".

Table 4 lists the datasets we used for our experiments. DataGov is an open-source government data portal containing thousands of government projects and their datasets. For our DataGov dataset, we crawled 1000 files and randomly selected 374 for our experiments. The number 374 was left after files were randomly shuffled and selected. Like DataGov, UKGov is also an open-source government portal containing UK government project files. For our UKGov dataset, we crawled 1578 project metadata files. We then manually selected 374 project metadata files and performed pre-processing before using them for experiments. Kaggle is an open-source portal aimed primarily at data science and machine learning experts. It contains datasets for data science and machine learning projects. For our Kaggle dataset, we crawled 563 project resources and manually selected 375 project files. Like the UKGov dataset, we also performed necessary pre-processing on this dataset before using it for our experiments. After DataGov's 374 files, numbers 374 and 375 were manually selected for UKGov and Kaggle, respectively, to avoid bias in MetaVader.

4.2 Performance Evaluation

The role of preparators and whether they improve metadata file readability, comprehensiveness, and reusability is the significant intuition behind our research work. Regarding this, we have designed and worked on experiments that analyze preparators and their role in improving metadata files. There are three main objectives for our experimentation according to the scope of this research paper (1) to understand and observe changes in metadata files before and after preparator applicability, (2) to understand and measure how each preparator is applicable on different files gathered from Kaggle, UKGov, and DataGov, and finally (3) to understand preparators and their performance. Our experiments aim to understand how preparators can improve the metadata content. Regarding quality issues prevalent in metadata files, we designed preparators that identify, resolve, and constitute metadata in various aspects, such as completion, changing existing data, extracting, and rectifying metadata content issues.

Concerning the scope of this research paper, we want to emphasize that the experiments conducted are based on the evaluation and usability of data preparators for metadata files. As far as our research experimentation is concerned, to the best of our knowledge, there are no substantial contributions to metadata preparation that directly utilize metadata or address metadata quality issues that can be resolved by applying preparation techniques. It is thus beyond the scope of this research paper to compare and evaluate our preparators on data files. Nevertheless, we aim to analyze the role of our preparators on structured and semi-structured files in our future work.

To understand how preparators behave and how important they are in detecting metadata elements, understanding metadata files, and identifying underlying issues, we designed an experiment to record the applicability of the preparator on all files in the MetaVader. Figure 4 shows the preparator applicability in percentages on the MetaVader. The most comprehensive and valuable observations from this experiment include (1) the role of preparators and applicability percentage and (2) identifying and understanding possible underlying issues in metadata files. For instance, the files obtained from DataGov were mainly structured with appropriate headers and details. The preparator applicability indicates the same pattern. However, the other two data sources comprised files with inconsistencies related to headers, misplaced metadata, irregular characters, empty header and columns, and missing values. Thus, preparator usability is higher in messy files.

We also performed the same applicability experiment after preparing the content of raw files to observe how the quality of the data changes after applying data preparators. Figure 5 shows the preparator applicability in percentages on the MetaVader after preparation. Similar to Figure 4, we can see the applicability on files from DataGov is low as the files from this source were structured compared to other sources. Overall, the preparators performed well, except for a few cases, for example, *standardize values* and *extract parts* where semantic understanding of the values poses a challenge in preparation.

We extended our experiment to observe how each preparator produces a change in metadata files. This experiment indicates changes observed in a metadata file after the application of preparators. To execute this experiment, we constructed a set of queries that would serve the purpose of questions that can be asked. These queries were applied to raw metadata files and prepared metadata files. The change in results was observed for each query, reflecting improvement in metadata content accessibility. In Table 5, we list seven queries amongst a group of other devised queries to explain the metrics we used to evaluate the raw and prepared metadata files. All queries are designed and equipped with search and extract methods. These methods can either search a file, look for content and return, i.e., extract the information or chunk from the file and display. For example, Q1 is designed to search and extract comments. The method is designed to analyze file contents and use *search* operation to identify and look for keywords or headers containing the word 'comments', 'comment', 'Comment', and its other possible variations. Similarly, get metadata type, i.e., administrative, structural, and descriptive queries, look for corresponding metadata elements, and if present, they are extracted and returned.

To further extend this experiment and evaluate the performance of MDPrep and the preparators, we used these designed sets of queries for both raw and prepared metadata files. Figure 6a and Figure 6b show the average score on the mentioned queries for raw and prepared metadata files, respectively. Figure 6a shows

Dataset Source	Total Files	Pre-Processing	Misplaced Metadata	MD Accessibility
Kaggle	375	91%	Yes, mostly	multiple files, text, comments, etc. (not in one place)
DataGov	374	35%	Partially	metadata inside files
UKGov	374	84%	Yes, mostly	dispersed, multiple files, text, comments, etc. (not in one place)

Table 4: The Table lists the datasets we used to develop and evaluate MDPrep and provides insights related to the amount in percentage for metadata pre-processing, misplaced metadata, and metadata accessibility.

Queries	Description	Example
Q1	Extract Comments	Look for header: comment(s), look for keywords: Relevance_list
Q2	Extract Hyperlinks	Search: Hyperlink (), Extract header (if applicable)
Q3	Missing Values	Search and Extract: rows, columns with empty spaces, long empty chunks
Q4	Count Headers	Count total headers in file and return title list
Q5	Get Descriptive MD	Extract descriptive metadata
Q6	Get Structural MD	Extract structural metadata
Q7	Get Administrative MD	Extract administrative metadata

Table 5: A list of evaluation queries applied to raw and prepared metadata files to observe changes rendered by preparators for performance comparison, usability, and reusability

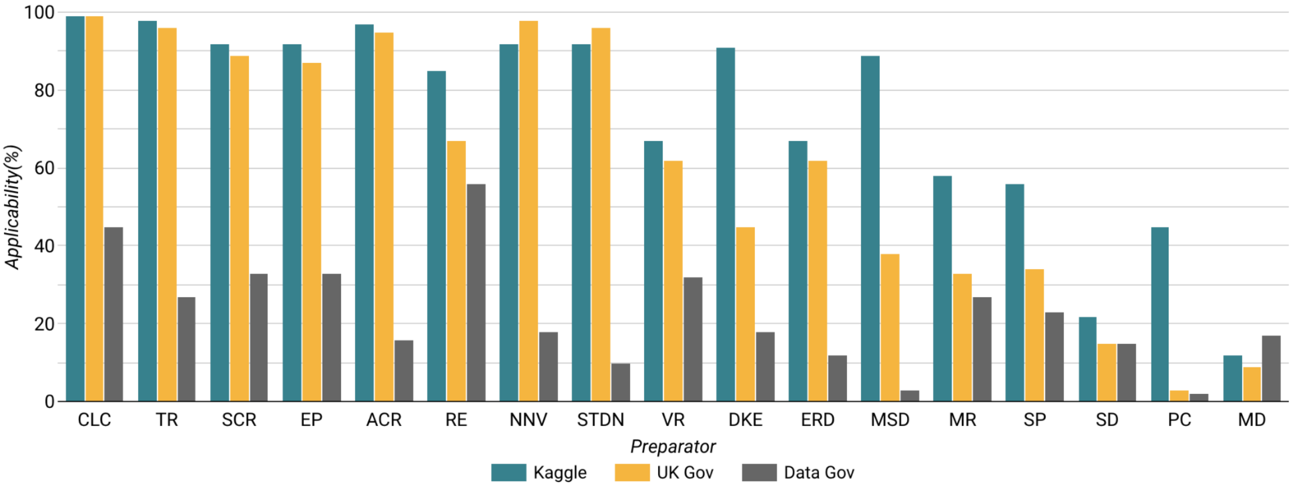


Figure 4: Data Preparators Applicability: this Figure depicts the role and measure of how applicable a preparator is on gathered (raw) files.

the performance of the queries we designed on raw metadata. It shows low applicability for Q5, Q6, and Q7 because the metadata for the raw files was missing, and if the metadata was present, it was not adequately prepared or labeled. On the other hand, since the files were prepared and appropriately labeled, we can see better applicability of these queries in Figure 6b. Not surprisingly, Q3's applicability to prepared metadata files is low when MDPrep used the preparator *Delete* instead of *Keep* for these files. Since MDPrep deletes the missing values and Q3 searches for missing data, there was no such data and, therefore, low applicability. For prepared metadata files, Q3 applicability is better when the user chooses *Keep* preparator in MDPrep setting.

It is critical to understand how preparators affect the overall quality of metadata files. We designed an experiment to evaluate the resolution of quality issues prevalent in metadata files after they have been amicably prepared. To comprehend preparators and their roles, we observed precision and recall measures to understand better how each preparator performs. Figure 7 depicts an overall view of each preparator.

Missing content is among the top-quality issues we observed for metadata files. MDPrep is designed to identify and highlight missing metadata in files. It does not compute missing values. However, many meaningful aspects can be addressed if an expert is added to this loop. For instance, if a metadata file is missing rights metadata, a prompt to an expert user can create the opportunity of addressing important information that is missing.

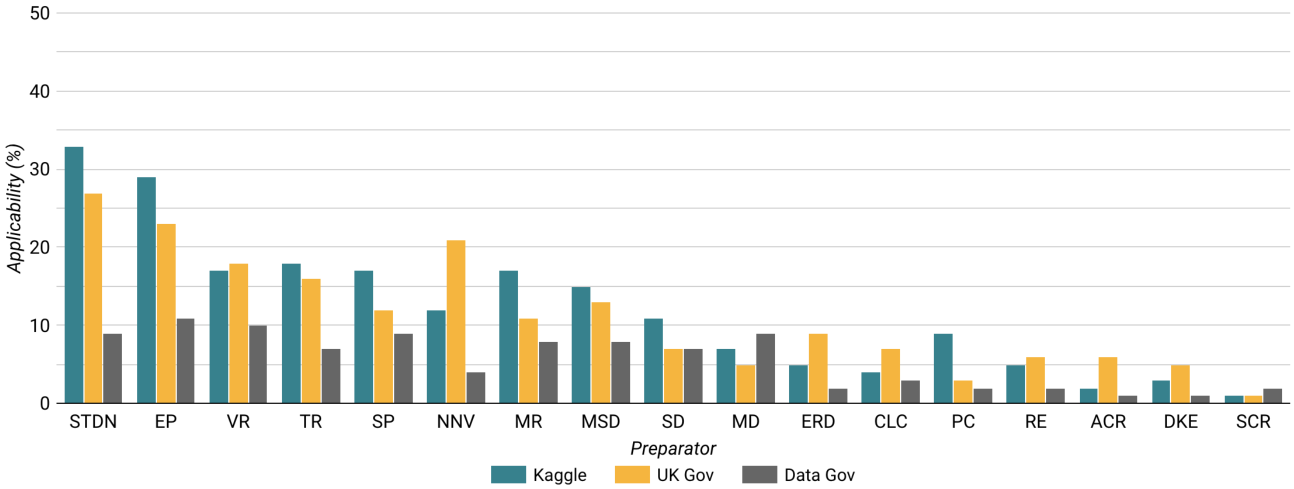
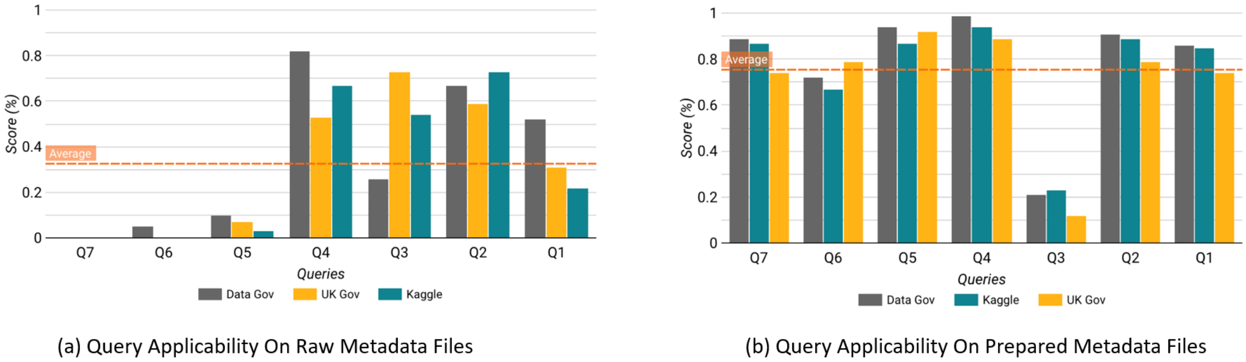


Figure 5: Data Preparators Applicability: this Figure depicts the role and measure of how applicable a preparator is on prepared files.



(a) Query Applicability On Raw Metadata Files

(b) Query Applicability On Prepared Metadata Files

Figure 6: Figure 6a shows the evaluation of queries on raw metadata files. It indicates how information is missing and unprepared in raw metadata files. The content is still detectable and comprehensible, but it is not classifiable. While Figure 6b shows the improvement in query applicability after the metadata files have been prepared. It improves content detection, classification, and comprehension and facilitates detecting different metadata types.

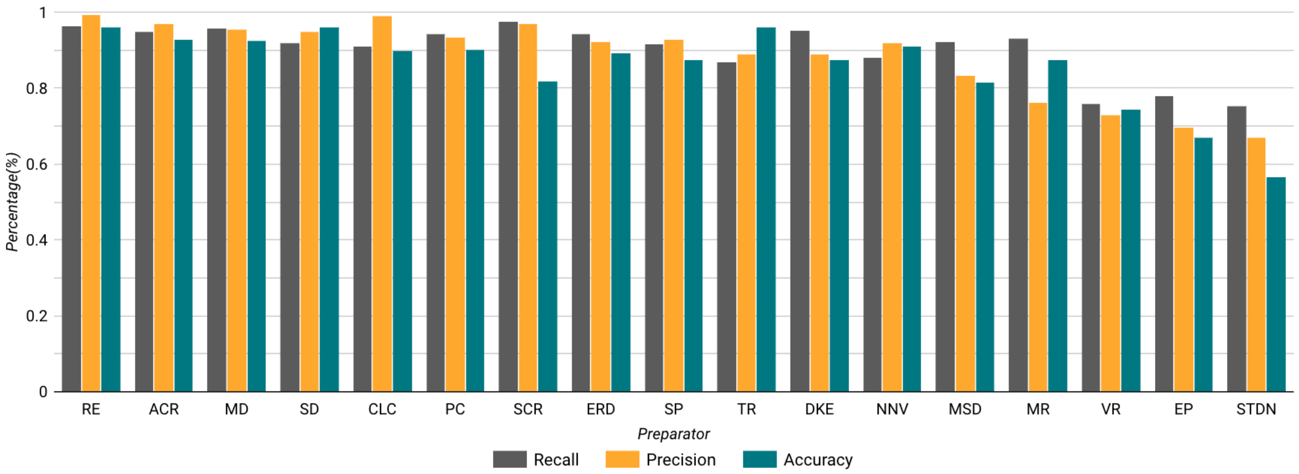


Figure 7: Preparator Performance: This Figure demonstrates precision, recall, and accuracy for each metadata preparator and its role in improving metadata file quality and identifying some major metadata content discrepancies.

5 RELATED WORK

As for the novelty of the research, to the best of our knowledge, there is no related work on data preparation techniques for metadata files and repositories. However, we have collected some notable research contributions that improve the data pre-processing by applying data preparation tasks.

Data Cleaning: Koumarelas et al. addressed data preparation for duplicate detection [18]. The authors expressed how algorithms could perform better in error detection when the utilized datasets are prepared. Hameed et al. addressed the problem of error detection in CSV files using syntactic patterns [10]. The authors presented how their approach can facilitate users to load their data by successfully identifying erroneous records. Kunft et al. use data preparation steps for efficient matrix partitioning through joins by separating the name by case, spaces, and other delimiters and hash the words into a fixed-size feature space [19]. Christodoulakis et al. proposed a rule-based approach to identify the semantic type of the data rows and classify them into different classes to clean the structure of CSV files [5].

Data Transformation: In cases where user input is possible by providing input/output examples, Zhongjun et el. proposed methods for synthesizing functions that learn how to perform these transformations and apply them to the remaining data [15]. The DataXFormer [1] is similarly based on the user-provided input/output examples but expects a relational table as input to transform it into a format defined by the examples. He et al. presented their system as an MS Excel plugin that proposes a set of options based on online repositories and programming scripts that the user can use to transform cell values [11]. Zhongjun et al. introduced the *transform-by-pattern* paradigm to transform cell values using patterns from online repositories and wiki tables [16].

6 CONCLUSION

This paper discussed the applicability and usability of data preparation techniques for raw, unprocessed metadata to determine changes and improvements. The research aimed to provide a basis for how data preparation techniques can extend and improve the reusability and management of metadata files. To achieve this, we experimented with a mixed collection of open data metadata files by applying preparators and observing significant changes in the file structure, file readability, and file management. Our experiments showed that file structure and readability improve after data preparation. We have also found that different files contain different information and quality issues that require custom preparators. This method of experimentation helped us understand the components of metadata files and how they can vary. We conducted the second experiment to observe how many preparators were applied to each metadata file. We concluded that our preparators could deal with various problems that metadata files contain. Finally, we show how the content of metadata changes through the use of preparators and how preparators improve the metadata quality.

REFERENCES

[1] Ziawasch Abedjan, John Morcos, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, and Michael Stonebraker. Dataxformer: A robust transformation discovery system. In *IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1134–1145. IEEE, 2016.

[2] Dan Brickley, Matthew Burgess, and Natasha Noy. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, pages 1365–1375, 2019.

[3] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. Auctus: a dataset search engine for data discovery and augmentation. *PVLDB*, 14(12):2791–2794, 2021.

[4] Adriane Chapman, Paolo Missier, Giulia Simonelli, and Riccardo Torlone. Capturing and querying fine-grained provenance of preprocessing pipelines in data science. *PVLDB*, 14(4):507–520, 2020.

[5] Christina Christodoulakis, Eric B Munson, Moshe Gabel, Angela Demke Brown, and Renée J Miller. Pytheas: pattern-based table discovery in csv files. *PVLDB*, 13(11):2075–2089, 2020.

[6] AnHai Doan, Pedro Domingos, and Alon Y Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 509–520, 2001.

[7] DublinCore. Dublin core metadata initiative.

[8] John S Erickson, Amar Viswanathan, Joshua Shinavier, Yongmei Shi, and James A Hendler. Open government data: a data analytics approach. *IEEE Intelligent Systems*, 28(5):19–23, 2013.

[9] Mazhar Hameed and Felix Naumann. Data preparation: A survey of commercial tools. *ACM SIGMOD Record*, 49(3):18–29, 2020.

[10] Mazhar Hameed, Gerardo Vitagliano, Lan Jiang, and Felix Naumann. Suragh: Syntactic pattern matching to identify ill-formed records. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, page 143–154, 2022.

[11] Yeye He, Xu Chu, Kris Ganjam, Yudian Zheng, Vivek Narasayya, and Surajit Chaudhuri. Transform-data-by-example (tde) an extensible search engine for data transformations. *PVLDB*, 11(10):1165–1177, 2018.

[12] Joseph M Hellerstein, Jeffrey Heer, and Sean Kandel. Self-service data preparation: Research to practice. 41(2):23–34, 2018.

[13] Mauricio A Hernández, Paolo Papotti, and Wang-Chiew Tan. Data exchange with data-metadata translations. *PVLDB*, 1(1):260–273, 2008.

[14] IEEE. Ieee lom: Ieee standard for learning object metadata.

[15] Zhongjun Jin, Michael R Anderson, Michael Cafarella, and HV Jagadish. Foofah: Transforming data by example. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD)*, pages 683–698, 2017.

[16] Zhongjun Jin, Yeye He, and Surajit Chauduri. Auto-transform: learning-to-transform by patterns. *PVLDB*, 13(12):2368–2381, 2020.

[17] Toralf Kirsten, Alexander Kiel, Mathias Rühle, and Jonas Wagner. Metadata management for data integration in medical sciences. *Datenbanksysteme für Business, Technologie und Web (BTW)*, 2017.

[18] Ioannis Koumarelas, Lan Jiang, and Felix Naumann. Data preparation for duplicate detection. *Journal of Data and Information Quality (JDIQ)*, 12(3):1–24, 2020.

[19] Andreas Kunft, Asterios Katsifodimos, Sebastian Schelter, Tilmann Rabl, and Volker Markl. Blockjoin: Efficient matrix partitioning through joins. *PVLDB*, 10(13):2061–2072, 2017.

[20] Sven Langenecker, Christoph Sturm, Christian Schalles, and Carsten Binnig. Towards learned metadata extraction for data lakes. *BTW 2021*, 2021.

[21] Yuliang Li, Xiaolan Wang, Zhengjie Miao, and Wang-Chiew Tan. Data augmentation for ml-driven data preparation and integration. *PVLDB*, 14(12):3182–3185, 2021.

[22] Jayant Madhavan, Philip A Bernstein, AnHai Doan, and Alon Halevy. Corpus-based schema matching. In *21st International Conference on Data Engineering (ICDE'05)*, pages 57–68. IEEE, 2005.

[23] Fatemeh Nargesian, Erkang Zhu, Renée J Miller, Ken Q Pu, and Patricia C Arocena. Data lake management: challenges and opportunities. *PVLDB*, 12(12):1986–1989, 2019.

[24] Felix Naumann and Melanie Herschel. An introduction to duplicate detection. *Synthesis Lectures on Data Management*, 2(1):1–87, 2010.

[25] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Q Pu, and Renée J Miller. Ronin: data lake exploration. *PVLDB*, 14(12):2863–2866, 2021.

[26] Tye Rattenbury, Joseph M Hellerstein, Jeffrey Heer, Sean Kandel, and Connor Carreras. *Principles of data wrangling: Practical techniques for data preparation.* O'Reilly Media, Inc., 2017.

[27] Franck Ravat and Yan Zhao. Metadata management for data lakes. In *European Conference on Advances in Databases and Information Systems*, pages 37–44. Springer, 2019.

[28] Sebastian Schelter, Joos-Hendrik Böse, Johannes Kirschnick, Thoralf Klein, and Stephan Seufert. Declarative metadata management: A missing piece in end-to-end machine learning. *Proceedings of SYSML*, 18, 2018.

[29] Gerrit JJ van den Burg, Alfredo Nazábal, and Charles Sutton. Wrangling messy csv files by detecting row and type patterns. *Data Mining and Knowledge Discovery*, 33(6):1799–1820, 2019.