Milestone 1 - Proof of Concept

Group member: Rong Huang, Xin Jiang, Junyu She

JSON Data Report

Introduction

This report describes three JSON datasets derived from different sources: CSV files, ONIX files, and MARC records. These datasets represent bibliographic and product metadata about books. The datasets have been processed to ensure data quality and consistency, and any issues encountered have been logged for review.

1. Dataset: mergedCSV.json

- **Source**: Two CSV files, CRWReportJob148737.xlsx and LibraryTitleCopyReportJob148738.xlsx.
- **Description**:
  - This dataset contains metadata about books such as Title, Material Type, Author, Publisher, Subjects, and Copies.
  - It combines information about the same books from the two CSV files, with LibraryTitleCopyReportJob148738 acting as supplementary data to enrich CRWReportJob148737.
- **Processing**:
  - The datasets were merged based on the book Title, with supplementary information such as Copies being appended from the second CSV.
  - Issues such as missing ISBN, Title, or Author were flagged in an issuesLog.
- **Use Cases**:
  - Suitable for library inventory management or creating a searchable database for books.
  - Provides basic bibliographic details along with information about the availability of copies.

2. Dataset: mergedONIX.json

- **Source**: Two ONIX files, LEEANDLOW_20210707.xml and Lerner_Print_ONIX_20240104104306.xml.
- **Description**:

  - This dataset provides rich bibliographic and commercial metadata for books, including Title, Author, ISBN, Publisher, Price, Subjects, and Audience Range.
  - It captures metadata for both print and electronic books, making it versatile for publishing and distribution.
- **Processing**:
  - ONIX XML files were parsed into JSON, and products were merged based on ISBN or Title.
  - Issues such as missing ISBN or Title, and conflicts in Author, Publisher, or Price were flagged in an issuesLog.
  - Duplicates were avoided, and missing fields were supplemented where possible.
- **Use Cases**:
  - Useful for publishers, retailers, or libraries to manage bibliographic records and pricing information.
  - Can be integrated into e-commerce platforms or digital catalogs.

3. Dataset: marc.json

- **Source**: MARC (Machine-Readable Cataloging) files, such as AuburnMiddleSchool.mrc.
- **Description**:
  - This dataset represents MARC bibliographic records, including Title, Publisher, Publication Year, Language, and control fields such as Record Control Number.
  - It provides detailed and structured information commonly used in library catalogs.
- **Processing**:
  - MARC files were parsed into JSON format using marc4js.
  - Identifiers were extracted and normalized, and any missing or conflicting data was logged in an issuesLog.
  - Records with incomplete or invalid MARC formatting were skipped, and error-handling mechanisms ensured smooth processing.
- **Use Cases**:
  - Ideal for library systems to import cataloging data.
  - Can support inter-library loan systems or metadata synchronization across institutions.

Data Processing Overview

- **Issue Logs**:
    - For each dataset, an issuesLog.json was generated to document data quality issues:
        - Missing essential fields like ISBN, Title, or Author.
        - Conflicts in bibliographic information such as Publisher or Price.
        - Duplicates within datasets.
    - These logs help ensure transparency and facilitate data correction or manual review.
- **Deduplication**:
    - Duplicates were identified and merged based on unique identifiers (e.g., ISBN or Title).
- **Data Enrichment**:
    - Missing fields in one dataset were supplemented using corresponding data from another.

Conclusion

These datasets collectively represent a comprehensive library of bibliographic metadata, useful for libraries, publishers, and retailers. By processing the data and addressing issues via logging, the datasets are now clean, enriched, and ready for integration into various systems, such as library catalogs, e-commerce platforms, or inventory management tools. Data are stored in the MongoDB database.

Answers to the Questions

1. **What is your prototype doing?**

Our prototype processes library data from MARC, CSV, and ONIX files, validates it, merges it, and stores it in MongoDB. The system ensures the data is clean, consistent, and ready for use in library management systems. It also detects and logs issues like missing fields or duplicate records.

2. **Can you enter in data?**

Yes, the system reads data from the files and enters it into MongoDB. For example:

- MARC records are inserted into the books_marc collection.
- CSV data is inserted into the books_csv collection.
- ONIX records are inserted into the books_onix collection.
- Any issues during processing are stored in issue log collections such as csvIssuesLog.

3. **Can you retrieve data?**

Yes, MongoDB allows us to retrieve data easily

4. **Do you have a front end interfacing with your database?**

Currently, there is no dedicated front end. I use MongoDB Atlas's built-in tools to visualize and query the data. However, this backend is ready to support a front end in the future. As a next step, we plan to develop a simple front-end interface where users can easily search for books, view detailed records, and interact with the database in a user-friendly way.