Historical Ledger OCR Project - Final Report

Project Overview

This project extracts structured financial data from 18th-19th century British parish ledgers using multimodal AI (GPT-4.1-mini). Scanned PDF documents containing handwritten accounting records are converted into machine-readable tabular data suitable for historical and economic research.

Date Range: 1704-1900
Total Documents: 33 PDF files
Total Pages: 271 pages
—

Executive Summary

The project underwent iterative development across three versions:

Version	Total Rows	Key Changes
V1 (Original)	7,344	Initial implementation with basic prompting
V2 (First Run)	7,123	Added confidence scoring, modular architecture
V2.1 (Final)	**7,533**	Enhanced prompt for titles, section headers, brace groupings

V2.1 achieves a 2.6% improvement over V1, extracting 189 additional rows while maintaining high confidence scores (0.963 average) and zero extraction errors.

Methodology

Data Pipeline Architecture

```
PDF Files → Image Conversion → Multimodal AI Extraction → Data Cleaning → Validation →
```

Technical Stack

- **AI Model:** GPT-4.1-mini (OpenAI)
- **PDF Processing:** PyMuPDF (fitz)
- **Image Handling:** Pillow (PIL)
- **Data Processing:** pandas
- **Output Format:** Excel (.xlsx)

Project Structure

```
ledger-ocr-project-v2/
├── src/
│   ├── config.py          # Settings and schema definitions
│   ├── pdf_utils.py       # PDF to image conversion
│   ├── schema.py          # Data cleaning and validation
│   ├── extraction.py      # AI extraction pipeline
│   └── validation.py      # Quality checks and comparison
├── data/                  # Input PDF files (33 ledgers)
├── outputs/               # Generated Excel files and logs
├── main.ipynb            # Main orchestration notebook
└── README.md
```

Key Features

1. Confidence Scoring Mechanism

Each extracted row receives a computed confidence score (0.0-1.0) based on: - Has description (+0.2) - Has at least one amount field (+0.2) - Valid pence fraction (+0.2) - Row type consistent with content (+0.2) - Amount fields are properly numeric (+0.2)

Results: Average confidence of 0.963 with zero low-confidence rows (<0.6).

2. Historical Notation Support

The system handles archaic British currency notations: - Unicode fractions: ¼, ½, ¾ - Historical abbreviations: “q” or “qd” = 1/4 (quarter pence), “ob” = 1/2 (half pence) - Denarius suffix: “3/4 d” → fraction is “3/4”

3. Row Type Classification

Four distinct row types are identified: - **title** (270 rows): Page headers with dates and document titles - **entry** (6,140 rows): Standard ledger entries with amounts - **section_header** (777 rows): Place names or labels without amounts - **total** (346 rows): Sum lines, often marked “Summa”

4. Brace Grouping Detection

The system detects curly brace `{` groupings where multiple sub-entries belong to one parent:

```
Tintinhull { Napper  — 02  5  5
           { Hopkins — 01 18  7
```

These are linked via the `group_brace_id` field.

5. Transaction Type Tagging

For balance sheet pages, entries can be tagged as: - credit / debit - income / expenditure

Version Evolution & Investigation

V1 → V2: Initial Refactoring

Changes made: - Modularized codebase into separate Python files - Added confidence scoring mechanism - Implemented transaction type field for balance sheets - Enhanced pence fraction handling with historical notations

Issue identified: V2 extracted 221 fewer rows than V1 (7,123 vs 7,344).

Investigation Process

We conducted systematic comparison between V1 and V2:

1. **Per-file comparison** revealed all 33 files had fewer rows in V2

2. **Page-level analysis** on file 1704 showed:
    - V1: 23 rows on Page 1
    - V2: 18 rows on Page 1
    - Missing: title rows, section headers
3. **Visual inspection** of source document confirmed:
    - Page titles were being skipped
    - Rows without amounts (section headers) were not captured
    - Brace groupings were not detected

V2 → V2.1: Prompt Enhancement

Root causes identified: 1. Prompt did not emphasize capturing title rows 2. Section headers (rows without amounts) were being skipped 3. No instructions for brace grouping detection

Prompt improvements: - Explicit instruction to ALWAYS capture page title as first row - Clear definition of section_header (rows with description but NO amounts) - Added brace grouping detection with `group_brace_id` field - Emphasized counting every visible line

Result: V2.1 extracts 7,533 rows — exceeding V1 by 189 rows (+2.6%).

Final Results

Row Extraction Summary

Metric	Value
Total Rows Extracted	7,533
Total Pages Processed	271
Total Files Processed	33
Average Confidence Score	0.963
Low Confidence Rows (<0.6)	0
Extraction Errors	0

Row Type Distribution

Row Type	Count	Percentage
entry	6,140	81.5%
section_header	777	10.3%
total	346	4.6%
title	270	3.6%

Page Type Distribution

Page Type	Count
ledger	7,385
balance_sheet	148

Data Schema (24 columns)

Column	Type	Description
file_id	string	PDF filename identifier
page_number	integer	Page within PDF (1-based)
page_type	string	“ledger” or “balance_sheet”
page_title	string	Title text at top of page
row_index	integer	Row order within page
row_type	string	entry/section_header/total/title
date_raw	string	Date if present
description	string	Item/place name
amount_pounds	string	£ value
amount_shillings	string	s value
amount_pence_whole	string	d whole value

Column	Type	Description
amount_pence_fraction	string	1/4, 1/2, 3/4, or empty
is_total_row	boolean	Flag for sum lines
group_brace_id	string	Links rows grouped by { brace
transaction_type	string	credit/debit/income/expenditure
num_col_1-6	string	Additional columns for complex layouts
confidence_score	float	Computed 0.0-1.0 quality score
entry_confidence	string	Categorical label
notes	string	Manual annotations

Known Limitations & Future Improvements

Current Limitations

1. **Amount-description misalignment**: Some total rows have amounts captured in description field (e.g., "£538/19/9")
2. **Transaction type coverage**: Only 107 of 7,533 rows have transaction_type populated
3. **Over-classification of section headers**: Some entries with amounts are incorrectly marked as section_header when the model fails to read the amounts

Recommended Future Improvements

1. **Two-pass extraction**: First pass for structure, second pass for amount verification
2. **Balance verification**: Automatically check if entry sums match total rows
3. **Field-level confidence**: Separate confidence scores for description vs amounts
4. **Full document context**: Process entire PDF at once (like colleague's Gemini approach) for better cross-page understanding

Conclusion

This project demonstrates the viability of using multimodal AI for digitizing historical handwritten ledgers. Through iterative development and systematic investigation, V2.1 achieves:

- **Higher accuracy** than the original implementation (+2.6% more rows)
- **Better structure recognition** (titles, section headers, brace groupings)
- **Robust confidence scoring** for quality assessment
- **Clean, modular codebase** for future extension

The resulting dataset of 7,533 structured rows from 271 pages across 196 years of parish records is suitable for quantitative historical analysis and further research.

Appendix: Files Delivered

1. `ledger_transcription_v2.1_latest.xlsx` — Final extracted dataset
2. `version_comparison_*.csv` — Version comparison metrics
3. `extraction_summary_*.csv` — Detailed extraction statistics
4. `src/` folder — Complete Python modules
5. `main.ipynb` — Orchestration notebook with full pipeline
6. `README.md` — Project documentation

Author: Hamid Ostadi
Date: December 2024
Supervisor: H-AI KHu Lab