Enriching Historical Records: An OCR and AI-Driven Approach for Database Integration

Zahra Abedi

Supervisors: Richard van Dijk & Gijs Wijnholds

Universiteit Leiden
The Netherlands

Index

- Introduction
- Data
- Methods
- Evaluation
- Discussion & Future Work
- Conclusion

Introduction

Background

Linking University, City and Diversity (LUCD)

- Visualize interactions between Leiden University and city of Leiden since 1575.

- Capture the impact of students and professors on Leiden.

- Collaborative work between researchers and students from LIACS and humanities faculty.

- A software system is designed which contains:
  - Database
  - Adapters for data extraction, transformation, loading and linking
  - Website for visualizing the results

Research Questions

Focus: Enriching the centralized database with “Leidse hoogleraren en lectoren 1575-1815” dataset.

Research Questions: How can we accurately extract and transform historical records data from scanned historical documents and map it into a centralized database?

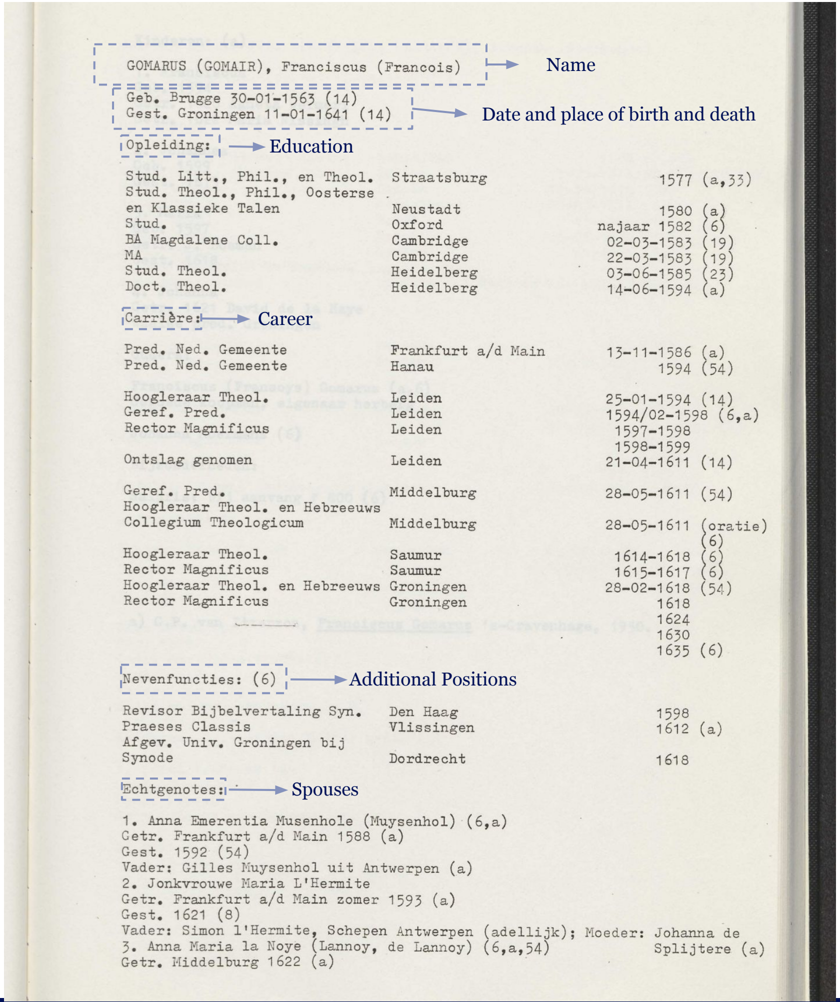
- SQ1: How can we extract high-accuracy text from scanned historical documents using OCR techniques?
- SQ2: How can AI play a role to analyze the OCR generated text and obtain a structured format?
- SQ3: How can we map the structured data into a centralized database?

Data

Data

Leidse hoogleraren en lectoren 1575-1815

- Compiled by A.A. Bantjes and L. van Poelgeest from 1983 to 1985
- Seven volumes
- Contains the following information about professors:
    - Date and place of birth and death
    - Education
    - Career history
    - Additional positions
    - Genealogical details regarding: Spouse(s), Children, Parents, Grandparents, etc.
    - Special details (salary, memberships, etc.)
    - Sources used


Name
Date and place of birth and death
Education
Career
Additional Positions
Spouses

GOMARUS (GOMAIR), Franciscus (Francois)

Geb. Brugge 30-01-1563 (14)
Gest. Groningen 11-01-1641 (14)

Opleiding:

Stud. Litt., Phil., en Theol.	Straatsburg	1577 (a,33)
Stud. Theol., Phil., Oosterse en Klassieke Talen	Neustadt	1580 (a)
Stud.	Oxford	najaar 1582 (6)
BA Magdalene Coll.	Cambridge	02-03-1583 (19)
MA	Cambridge	22-03-1583 (19)
Stud. Theol.	Heidelberg	03-06-1585 (23)
Doct. Theol.	Heidelberg	14-06-1594 (a)

Carrière:

Pred. Ned. Gemeente	Frankfurt a/d Main	13-11-1586 (a)
Pred. Ned. Gemeente	Hanau	1594 (54)
Hoogleraar Theol.	Leiden	25-01-1594 (14)
Geref. Pred.	Leiden	1594/02-1598 (6,a)
Rector Magnificus	Leiden	1597-1598 1598-1599
Ontslag genomen	Leiden	21-04-1611 (14)
Geref. Pred.	Middelburg	28-05-1611 (54)
Hoogleraar Theol. en Hebreeuws Collegium Theologicum	Middelburg	28-05-1611 (oratie) (6)
Hoogleraar Theol.	Saumur	1614-1618 (6)
Rector Magnificus	Saumur	1615-1617 (6)
Hoogleraar Theol. en Hebreeuws	Groningen	28-02-1618 (54)
Rector Magnificus	Groningen	1618 1624 1630 1635 (6)

Nevenfuncties: (6)

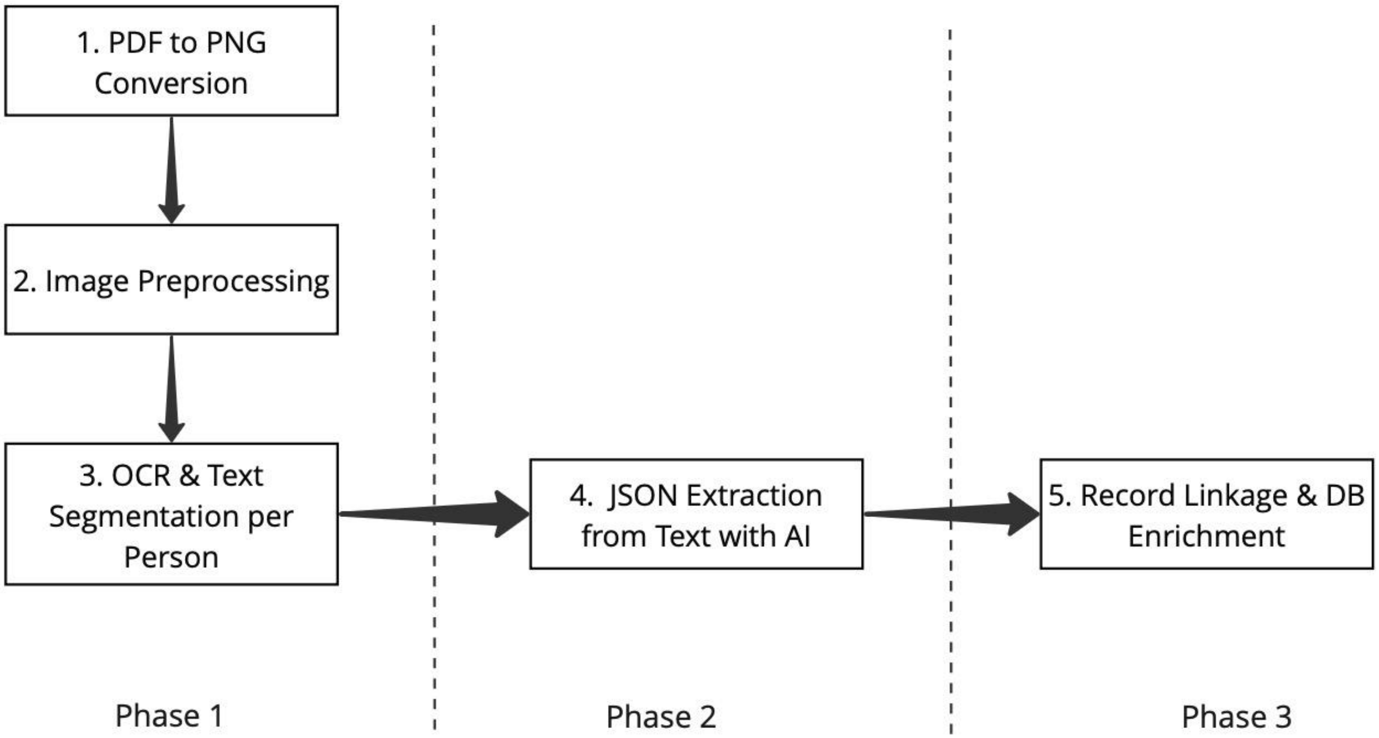
Revisor Bijbelvertaling Syn.	Den Haag	1598
Praeses Classis	Vlissingen	1612 (a)
Afgev. Univ. Groningen bij Synode	Dordrecht	1618

Echtgenotes:

1. Anna Emerentia Musenhole (Muysenhol) (6,a)
Getr. Frankfurt a/d Main 1588 (a)
Gest. 1592 (54)
Vader: Gilles Muysenhol uit Antwerpen (a)
2. Jonkvrouwe Maria L'Hermite
Getr. Frankfurt a/d Main zomer 1593 (a)
Gest. 1621 (8)
Vader: Simon l'Hermite, Schepen Antwerpen (adellijk); Moeder: Johanna de Splijtere (a)
3. Anna Maria la Noye (Lannoy, de Lannoy) (6,a,54)
Getr. Middelburg 1622 (a)

Methods

Methods



1. PDF to PNG Conversion

2. Image Preprocessing

3. OCR & Text Segmentation per Person

4. JSON Extraction from Text with AI

5. Record Linkage & DB Enrichment

Methods

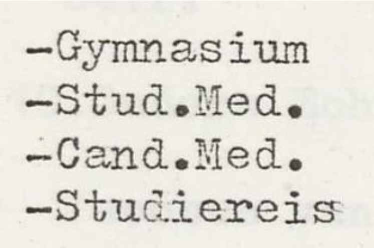
Phase 1

1. PDF to PNG Conversion

- Chose PNG for its balance of quality and file size.
- Necessary to convert PDFs to a compatible format for Tesseract OCR

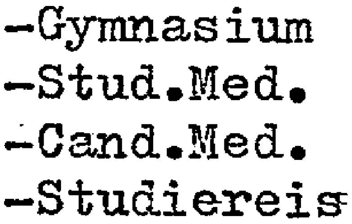
2. Image Preprocessing Using OpenCV (cv2)

- Image Denoising
- Conversion to Grayscale
- Grayscale to Binary Conversion



-Gymnasium
-Stud.Med.
-Cand.Med.
-Studiereis

Before Preprocessing



-Gymnasium
-Stud.Med.
-Cand.Med.
-Studiereis

After Preprocessing

Methods

Phase 1

3. Optical Character Recognition (OCR) & Text Segmentation per Person

OCR:

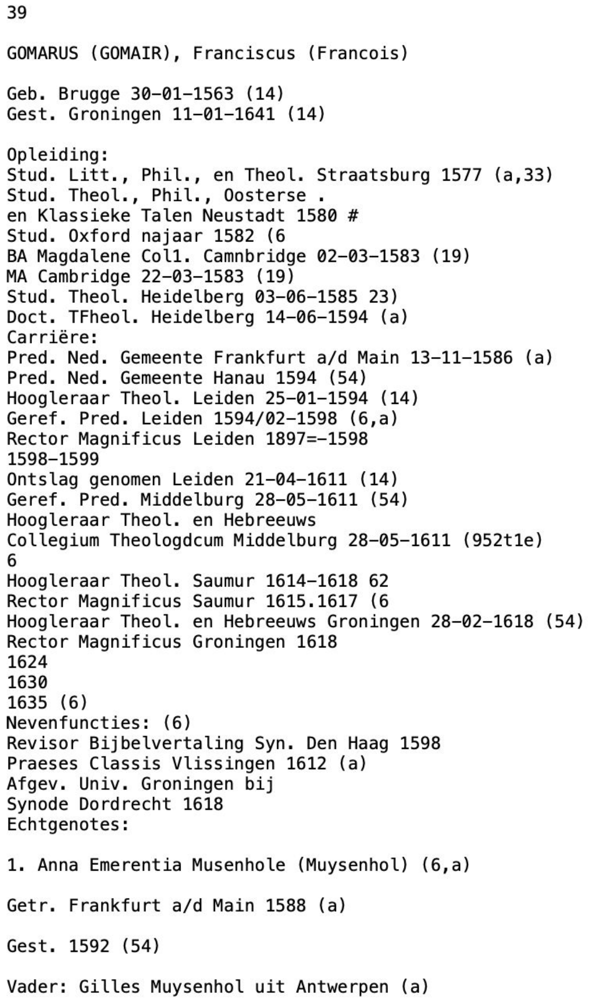
- Tool Selection: Tesseract
- Language configuration set to Dutch
- Training Tesseract
- Word List Integration
- Page Segmentation Modes (PSM)

Methods

Phase 1

Text Segmentation per Person

- Split the text per person using Regular Expressions (regex) in Python.
  - Identify last names written in all caps. (e.g., GOMARUS)
  - Look for strings with 3+ consecutive capital letters



```
39

GOMARUS (GOMAIR), Franciscus (Francois)

Geb. Brugge 30–01–1563 (14)
Gest. Groningen 11–01–1641 (14)

Opleiding:
Stud. Litt., Phil., en Theol. Straatsburg 1577 (a,33)
Stud. Theol., Phil., Oosterse .
en Klassieke Talen Neustadt 1580 #
Stud. Oxford najaar 1582 (6
BA Magdalene Col1. Camnbridge 02–03–1583 (19)
MA Cambridge 22–03–1583 (19)
Stud. Theol. Heidelberg 03–06–1585 23)
Doct. TFheol. Heidelberg 14–06–1594 (a)
Carriëre:
Pred. Ned. Gemeente Frankfurt a/d Main 13–11–1586 (a)
Pred. Ned. Gemeente Hanau 1594 (54)
Hoogleraar Theol. Leiden 25–01–1594 (14)
Geref. Pred. Leiden 1594/02–1598 (6,a)
Rector Magnificus Leiden 1897=–1598
1598–1599
Ontslag genomen Leiden 21–04–1611 (14)
Geref. Pred. Middelburg 28–05–1611 (54)
Hoogleraar Theol. en Hebreeuws
Collegium Theologdcum Middelburg 28–05–1611 (952t1e)
6
Hoogleraar Theol. Saumur 1614–1618 62
Rector Magnificus Saumur 1615.1617 (6
Hoogleraar Theol. en Hebreeuws Groningen 28–02–1618 (54)
Rector Magnificus Groningen 1618
1624
1630
1635 (6)
Nevenfuncties: (6)
Revisor Bijbelvertaling Syn. Den Haag 1598
Praeses Classis Vlissingen 1612 (a)
Afgev. Univ. Groningen bij
Synode Dordrecht 1618
Echtgenotes:

1. Anna Emerentia Musenhole (Muysenhol) (6,a)

Getr. Frankfurt a/d Main 1588 (a)

Gest. 1592 (54)

Vader: Gilles Muysenhol uit Antwerpen (a)
```

Methods

Phase 2

4. JSON Extraction from Text with AI

- Extract relevant information from text files into structured JSON format.
- Tools Used:
  - Pydantic for schema definition and data validation.
  - GPT-3.5 Turbo for data extraction.

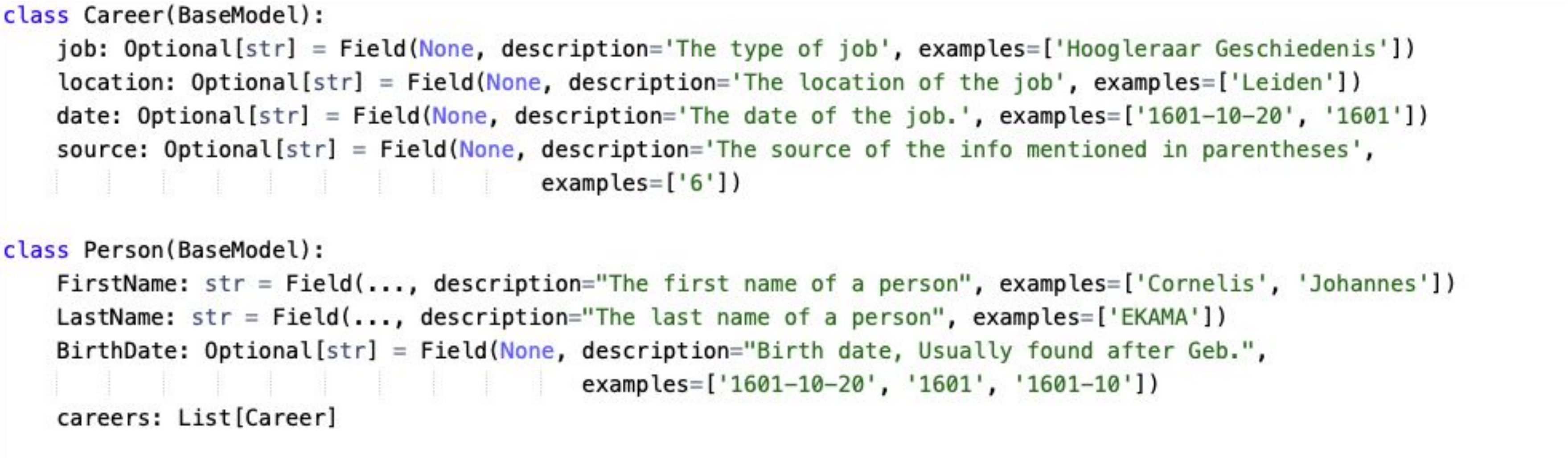
Pydantic

Methods

Phase 2

Schema Definition Using Pydantic

- Generate consistent output with all necessary fields.
- Example Pydantic Class:



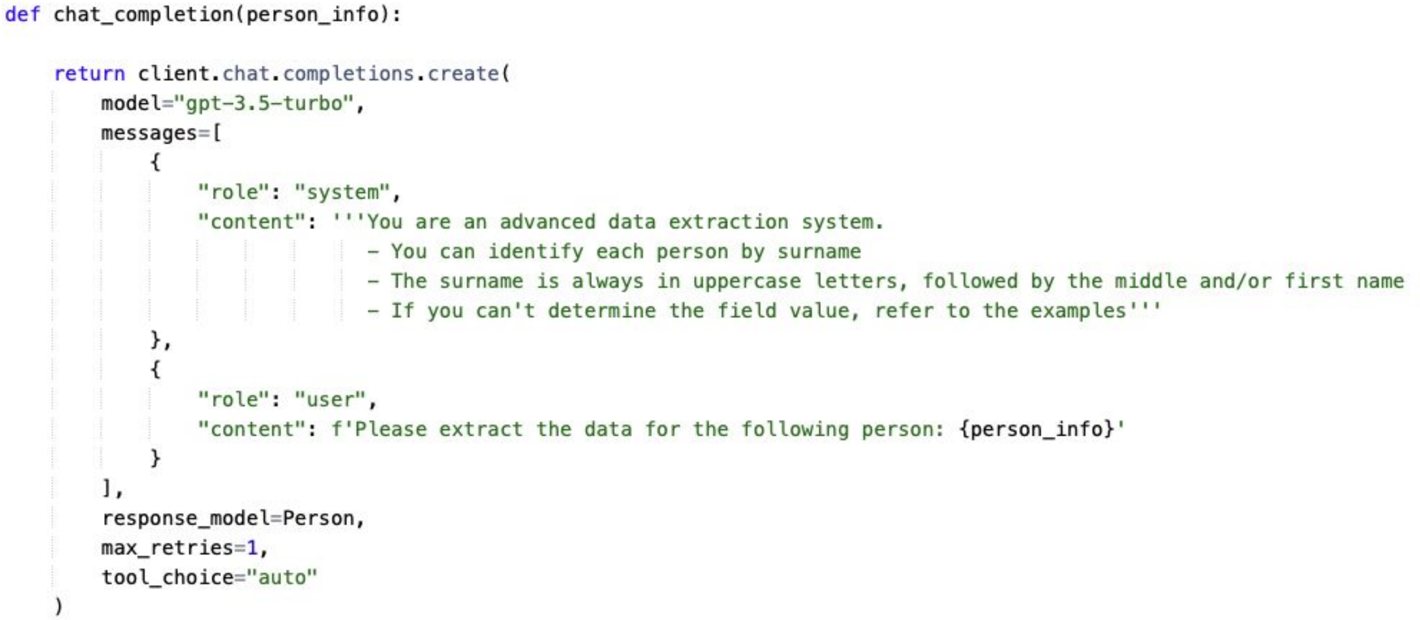
```
class Career(BaseModel):
    job: Optional[str] = Field(None, description='The type of job', examples=['Hoogleraar Geschiedenis'])
    location: Optional[str] = Field(None, description='The location of the job', examples=['Leiden'])
    date: Optional[str] = Field(None, description='The date of the job.', examples=['1601-10-20', '1601'])
    source: Optional[str] = Field(None, description='The source of the info mentioned in parentheses',
                                        examples=['6'])

class Person(BaseModel):
    FirstName: str = Field(..., description="The first name of a person", examples=['Cornelis', 'Johannes'])
    LastName: str = Field(..., description="The last name of a person", examples=['EKAMA'])
    BirthDate: Optional[str] = Field(None, description="Birth date, Usually found after Geb.",
                                            examples=['1601-10-20', '1601', '1601-10'])
    careers: List[Career]
```

Methods

Phase 2

Extraction Techniques Using GPT-3.5 Turbo

- Function Calling: Ensures the AI consistently generates valid JSON outputs according to predefined schema.
- GPT Prompt Used:



```
def chat_completion(person_info):

    return client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[
            {
                "role": "system",
                "content": '''You are an advanced data extraction system.
                              - You can identify each person by surname
                              - The surname is always in uppercase letters, followed by the middle and/or first name
                              - If you can't determine the field value, refer to the examples'''
            },
            {
                "role": "user",
                "content": f'Please extract the data for the following person: {person_info}'
            }
        ],
        response_model=Person,
        max_retries=1,
        tool_choice="auto"
    )
```

Methods

Phase 3

5. Record Linkage & Database Enrichment

- Enrich the centralized database developed by the LUCD project with data from JSON files.
- New tables and columns added
- Rating system to differentiate data quality:
  - Rating 3: High quality original data
  - Rating 2: Additional data matches existing entity
  - Rating 1: Entirely new entities

Methods

Phase 3

Linking Algorithm

- Partial matches to ensure flexibility
  - Example: 'Casper Janszoon' and 'Casper Johannes' considered a match
- Linking records based on specific conditions:

  First condition:
  - First name and last name match
  - Birth year or birth city matches

  Second condition:
  - Last name matches
  - Birth year matches
  - Birth city or birth country matches
- Handling uncertain matches:
  - Names match, but birth year and birthplace do not
  - Create a new person with a relation to the potentially matching individual

Evaluation

Evaluation

General Evaluation Approach:

- Sample comprising 10% of the total number of individuals from our dataset (40 individuals)
- Assessment of Each Phase:
    - Phase 1 Evaluation: Quality Assessment of Generated Text
    - Phase 2 Evaluation: Quality Assessment of Generated JSON
    - Phase 3 Evaluation: Quality Assessment of Linking Algorithm

Evaluation

Phase 1 Evaluation: Quality Assessment of Generated Text

- Ground Truth: 40 manually created .txt files
- Metrics used: Character Error Rate (CER) and Word Error Rate (WER)

$$\mathrm{WER} = \frac{S_w + D_w + I_w}{N_w}$$

- Sc is the number of word substitutions,
- Dc is the number of word deletions,
- Ic is the number of word insertions,
- Nc is the total number of words in the reference.

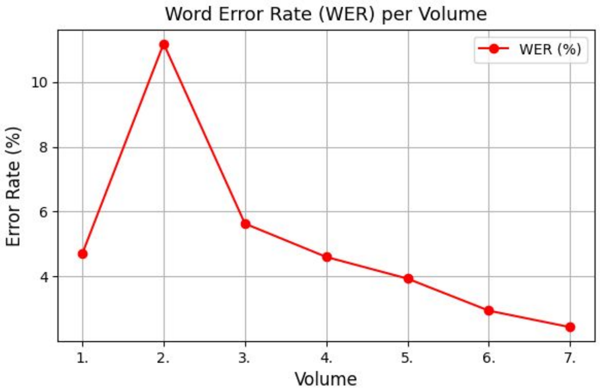
$$\mathrm{CER} = \frac{S_c + D_c + I_c}{N_c}$$

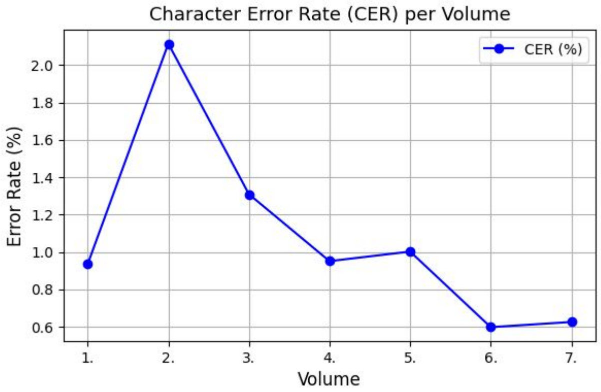
- Sc is the number of character substitutions,
- Dc is the number of character deletions,
- Ic is the number of character insertions,
- Nc is the total number of characters in the reference.

Evaluation

Phase 1 Evaluation: Quality Assessment of Generated Text

- Comparison of Average WER and CER per Volume:
  - WER: Higher due to word-level error accumulation.
  - CER: Lower, reflecting character-level errors.
- Volume 2 shows significantly higher error rates, likely due to poor print quality (faded ink, smudges).





Evaluation

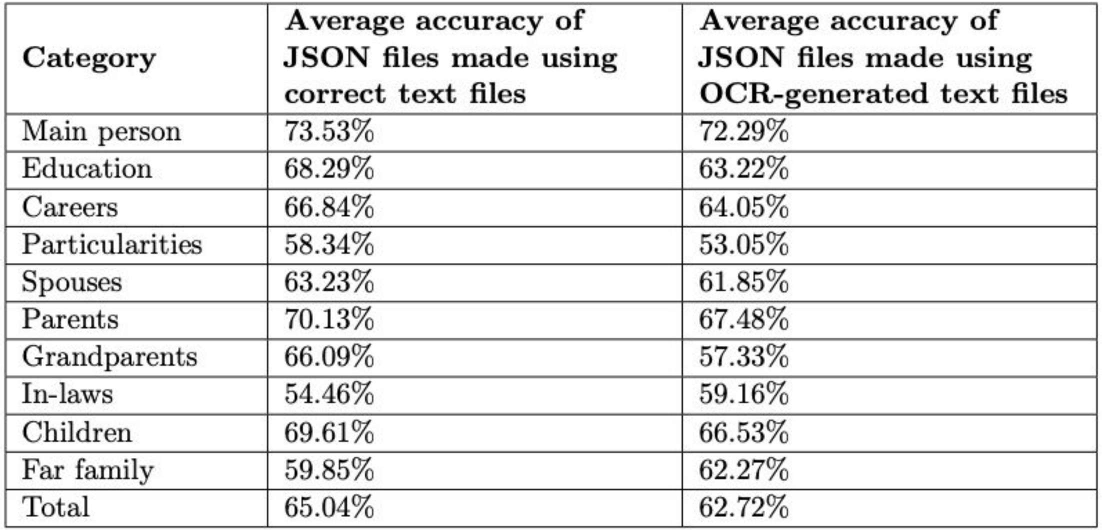
Phase 2 Evaluation: Quality Assessment of Generated JSON

- Ground Truth: 40 manually created JSON files
- Evaluation Sets:
  - Set 1: JSON files created using manually corrected text inputs.
  - Set 2: JSON files created using OCR-generated text inputs.
- Metrics and Methodology:
  - Normalization by lowercasing.
  - Key-value pair comparison.
  - Accuracy assessment.
  - Key categorization (e.g. 'Main person', 'Education', 'Careers', etc.)

Evaluation

Phase 2 Evaluation: Quality Assessment of Generated JSON

- Table of overall accuracy results per category.
- Comparison between JSON files from correct text files and OCR-generated text files.

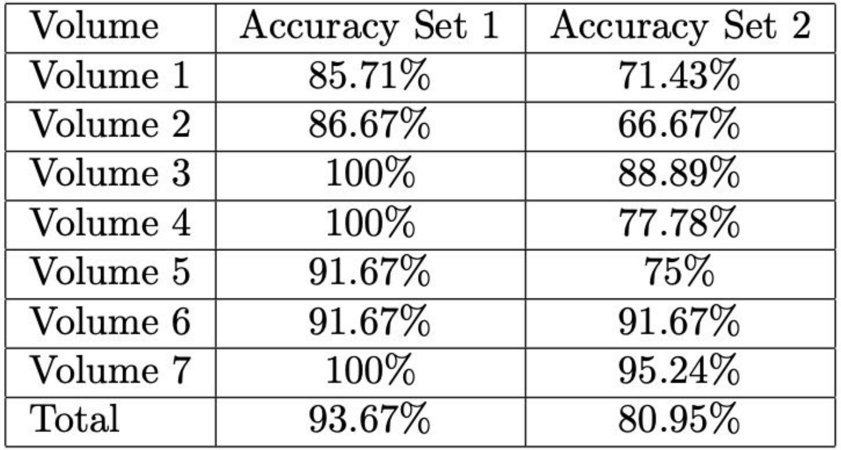


Category	Average accuracy of JSON files made using correct text files	Average accuracy of JSON files made using OCR-generated text files
Main person	73.53%	72.29%
Education	68.29%	63.22%
Careers	66.84%	64.05%
Particularities	58.34%	53.05%
Spouses	63.23%	61.85%
Parents	70.13%	67.48%
Grandparents	66.09%	57.33%
In-laws	54.46%	59.16%
Children	69.61%	66.53%
Far family	59.85%	62.27%
Total	65.04%	62.72%

Evaluation

Phase 3 Evaluation: Quality Assessment of Linking Algorithm

- Evaluate the performance of the enrichment algorithm on two sets of JSON files:
  - Set 1: Manually created JSON files.
  - Set 2: JSON files created using OCR-generated text inputs.



Volume	Accuracy Set 1	Accuracy Set 2
Volume 1	85.71%	71.43%
Volume 2	86.67%	66.67%
Volume 3	100%	88.89%
Volume 4	100%	77.78%
Volume 5	91.67%	75%
Volume 6	91.67%	91.67%
Volume 7	100%	95.24%
Total	93.67%	80.95%

Discussion & Future Work

Discussion & Future Work

Discussion:

- Residual OCR issues affecting downstream tasks
- Consider adding frequently appearing details to JSON
- Sample size in evaluation limited for practical reasons

Future Work:

- Advanced AI Models
- Improved Linking Algorithm
- Prompt Engineering

Conclusion

Conclusion

Research Question:

- How can we accurately extract and transform historical records data from scanned historical documents and map it into a centralized database?

Three-Phase Methodology:

- Phase 1: Text Extraction from PDFs using OCR
- Phase 2: JSON Extraction from Text with AI
- Phase 3: Record Linkage & Database Enrichment

Achievements:

- Enhanced OCR accuracy through preprocessing
- Structured data using Pydantic and OpenAI's GPT-3.5 Turbo
- Modified database schema and developed a record linking algorithm

Thank You!

Universiteit Leiden
The Netherlands