MedDocParser: Automated OCR, Medical NER, and ICD-10 Coding

BTech Project

Aatif Ahmad

BTech in AI and Data Science

IIT Jodhpur

17th November, 2025

Supervisors:

1. **Sumit Kalra**
Assistant Professor
Department of Computer Science and Engineering
IIT Jodhpur

2. **Manik Sejwal**
PhD in Medical Technologies
IIT Jodhpur

Abstract

MedDoc-Parser is a prototype for automated parsing of medical documents like prescriptions. It uses OCR for text extraction, NLP for identifying entities such as diseases, symptoms, and medications, and maps them to ICD-10 codes. The system is deployed via a Streamlit web interface for easy uploading and viewing of results.

This project tackles healthcare data digitization by minimizing manual errors and aiding EHR integration. Tests on sample documents show effective entity extraction and code mapping. Limitations include API dependencies and challenges with handwritten text. Future work involves local models and expanded ICD coverage.

Keywords: Medical Document Parsing, OCR, NLP, ICD-10

Contents

1 Introduction

1.1 Background

Healthcare generates unstructured data from prescriptions and reports. Manual entity extraction is inefficient and error-prone, complicating EHR management and ICD-10 standardization.

MedDoc-Parser automates this using AI: OCR for digitization, NLP for entities, and mapping for codes. This project applies computer vision and NLP to healthcare.

1.2 Objectives

- Build an end-to-end parsing system for medical images and PDFs.
- Extract entities with LLMs.
- Map to ICD-10 codes accurately.
- Create a simple web interface.

1.3 Scope and Limitations

Focuses on English printed text, using a subset of ICD-10 codes. API reliance raises privacy issues; needs compliance for production.

2 Literature Review

Tools like Google's Vision API handle OCR, while spaCy or BERT manage medical NER. ICD mapping uses rules in cTAKES or transformers in BioBERT.

This project employs PaddleOCR for extraction and GPT-4o-mini for flexible NER, with fuzzy matching to improve recall on medical terms.

3 Methodology

3.1 System Architecture

The pipeline has three stages:

1. OCR: Text extraction with PaddleOCR.
2. NER: Entity identification via GPT-4o-mini.
3. ICD Mapping: Matching using exact, substring, and fuzzy methods.

App.py orchestrates, with Streamlit for the UI.

3.2 OCR Module

PaddleOCR detects and extracts text, including rotated elements, from documents.

3.3 NER Module

Prompts GPT-4o-mini to output JSON with diseases, symptoms, medications, lab values, and notes.

3.4 ICD Mapping

Loads a CSV of ICD codes and matches entities through multiple levels for robustness.

4 Implementation

Built in Python with Streamlit, PaddleOCR, and OpenAI. Uses .env for keys. Key components: `app_ui.py` for UI, `app.py` for processing, `icd10.csv` for codes.

Run via streamlit run `app_ui.py` for demo.

5 Results and Discussion

5.1 Evaluation

Tested on sample prescriptions, showing reliable extraction of entities and mapping to relevant ICD codes, like "hypertension" to I10.

5.2 Discussion

The design supports scaling. It outperforms basic tools in adaptability without custom training.

6 Conclusion and Future Work

MedDoc-Parser prototypes AI for medical parsing, meeting healthcare automation goals. It excels in modularity but can improve with offline processing.

Future improvements include:

- Use local transformers for NER.
- Full ICD dataset integration.
- UI enhancements for entity highlighting.
- Real-data validation.

References

1. PaddleOCR Documentation. `https://github.com/PaddlePaddle/PaddleOCR`.

2. OpenAI API. `https://platform.openai.com/docs`.

3. WHO ICD-10. `https://icd.who.int`.

4. Shi et al., "BioBERT: a pre-trained biomedical language representation model." *Bioinformatics*, 2019.