📘 Technical Documentation

Layout-Aware Structured Extraction for Hospital Bills

1️⃣ Problem Framing

The objective was to extract structured fields from heterogeneous Indian hospital billing documents containing:

- Variable layouts
- Multi-page invoices
- Tabular line-item sections
- Insurance blocks
- Mixed typography
- Scanned + digitally generated PDFs

Traditional rule-based extraction approaches fail under such variability. Therefore, a layout-aware transformer-based approach was adopted.

2️⃣ System Architecture Overview

The pipeline consists of:

1. OCR (Tesseract)
2. Spatial token alignment (bounding box normalization)
3. LayoutLMv3 fine-tuned as Question Answering (span prediction)
4. Structured JSON generation

Instead of token classification (BIO tagging), the task was reformulated as document-level Question Answering.

Each field is extracted via a natural-language query:

- "What is the hospital name?"
- "What is the bill date?"
- "What is the discharge date?"
- etc.

This transforms structured extraction into span prediction over layout-aware token embeddings.

3️⃣ Technical Hurdles Faced

🔹 3.1 Delayed Dataset Access

Due to NDA-based gated access, dataset availability was delayed.
This reduced available training time and constrained experimentation cycles.

🔹 3.2 Absence of Ground Truth Annotations

The dataset did not include structured annotations.

Therefore:

- A schema had to be designed manually.
- Ground truth spans had to be generated.
- OCR-token alignment had to be validated.

This required careful synchronization between:

- OCR output tokens
- Annotation spans
- Bounding box indices

Annotation–token misalignment initially caused label collapse.

🔹 3.3 Token Classification Collapse

Initial experiments used BIO tagging with `LayoutLMv3ForTokenClassification`.

However, due to:

- Sparse entity tokens
- Severe class imbalance (majority "O")
- Limited annotated samples (~30 documents)

The model collapsed into predicting only background labels.

This is a known issue in low-data NER setups.

🔹 3.4 Reformulation to QA-Based Span Prediction

To address class imbalance and sparse supervision:

The task was reframed as Question Answering.

Advantages:

- Denser supervision signal
- More stable loss landscape
- Better utilization of limited annotations
- Reduced background-class dominance

This significantly stabilized training.

🔹 3.5 OCR Spatial Normalization

LayoutLMv3 requires bounding boxes normalized to 0–1000 scale.

Feeding raw pixel coordinates leads to spatial embedding mismatch and degraded predictions.

Bounding box normalization was critical for correct inference.

4️⃣Why Only 3 Training Epochs?

Training was limited to 3 epochs due to:

- Time constraints after dataset access delay
- GPU runtime limitations
- Small dataset size (76 QA samples)

Loss curve progression:

- Epoch 1: 5.14
- Epoch 2: 3.42
- Epoch 3: 2.78

The model was still converging. Additional epochs would likely improve Exact Match and F1.

Given limited supervision, longer training risks overfitting. Therefore, training was stopped conservatively.

5️⃣Why Some Documents Return All Null Values

Some documents may output null values due to:

1. OCR inconsistencies (poor scan quality)
2. Field not present in document
3. Span prediction confidence low
4. Layout variation not sufficiently represented in training set
5. Token mismatch between OCR at training and inference time

Since extraction is learned (not rule-based), the model predicts spans only when confident.

No hardcoded fallbacks were used to artificially inflate accuracy.

6️⃣ Why Regex-Based Extraction Is Not Used

Regex-based systems are intentionally avoided due to:

- High template variability across hospitals
- Different date formats
- Different placement of MRN, Bill No., Insurance fields
- Multi-column layouts
- Tabular ambiguity
- Multi-page continuation formats

Regex approaches:

- Break when field order changes
- Fail when labels are missing
- Cannot reason about layout
- Cannot generalize to unseen templates

In contrast, LayoutLMv3:

- Uses spatial embeddings
- Learns layout relationships
- Handles structural variation
- Generalizes better across templates

Thus, a learning-based approach was chosen for robustness and scalability.

7️⃣ How LayoutLM Handles Variable-Length Tables

The concern raised in the demo call:

> "Tabular data varies — some bills have 2 line items, others have 10+. How does annotation scale?"

Answer:

LayoutLM does not require fixed-length structures.

It operates at the token level with spatial embeddings.

For tabular extraction:

- Each row is just a sequence of spatially related tokens.
- The model learns row-wise spatial clustering.
- Variable-length tables are naturally handled because transformers operate over sequences, not fixed grids.

In the current implementation, structured header fields are prioritized.
Line-item extraction can be extended via:

- Row-level grouping logic
- Span prediction per table column
- Layout-aware region classification

Thus, LayoutLM is inherently capable of handling variable-length tables without fixed schema constraints.

8️⃣ Image-Only Demo Clarification

The current demo accepts:

- JPG
- PNG
- Image-based inputs

However, it can be extended to support:

- Scanned PDFs (convert PDF → image pages)

- Digital PDFs (via pdf2image)

Preprocessing pipeline already supports image conversion; extension is straightforward.

9️⃣ Current Limitations

- Small annotated dataset
- Limited training epochs
- No held-out test evaluation
- OCR dependency
- No confidence scoring per field yet

🔟 Scalability Roadmap

With more data:

- Increase QA samples
- Add table-level extraction
- Introduce field confidence thresholds
- Fine-tune longer
- Explore domain-adaptive pretraining

Conclusion

Despite limited supervision and heterogeneous templates, the QA-based LayoutLMv3 approach demonstrates:

- Stable training

- Layout-aware reasoning
- Structured field extraction capability
- Scalability potential

This solution avoids brittle rule-based heuristics and provides a principled transformer-based document intelligence framework.