

Trích xuất dữ liệu từ tài liệu

Các phương pháp từ truyền thống đến hiện đại

Chủ đề: Document Parsing

Nhóm trình bày:

22120252 - Giang Đức Nhật

22120427 - Nguyễn Mạnh Văn

22120450 - Bùi Đình Gia Vỹ

1. Bài toán thực tế

Hãy tưởng tượng chúng ta xây dựng ứng dụng theo dõi chi tiêu cho một công ty. Nhân viên tải lên hàng trăm hóa đơn mỗi ngày.

Mục tiêu: Tự động trích xuất **Ngày tháng**, **Tổng tiền**, và **Danh sách mặt hàng**.

Chúng ta thường gặp 2 loại file chính:

1. **Digital-Native PDFs:** Xuất trực tiếp từ Word/Excel. Thẳng hàng, sạch đẹp.
2. **Scanned/Image PDFs:** Ảnh chụp bằng điện thoại hoặc máy scan cũ. Mờ, nghiêng, nhiều.

Vậy với mỗi loại file này chúng ta có những cách nào để giải quyết?

DEMO TIME (PHẦN 1)

Cách tiếp cận "Truyền thống"

Kịch bản: Chúng ta sẽ thử đọc một **Hóa đơn Scan** bằng thư viện `pdfplumber`.

(Chuyển sang màn hình Streamlit)

2. Tại sao nó thất bại? (Nguyên lý)

Công cụ chúng ta vừa dùng (`pdfplumber`) hoạt động dựa trên **Tọa độ (Coordinate-based)**.

Cơ chế hoạt động:

Nó KHÔNG "nhìn" thấy tài liệu như mắt người. Nó đọc **Cây đối tượng PDF (PDF Object Tree)**.

- Nó tìm các lệnh như: `Vẽ chữ "Total" tại x=100, y=500` .

Hạn chế chết người:

- Với file Scan (Ảnh), **KHÔNG HỀ CÓ lệnh vẽ chữ nào**.
- Đối với `pdfplumber` , file scan chỉ là một trang giấy trắng chứa một bức ảnh to.
- Ngay cả với file Digital, nếu bố cục bị lệch một chút, code tọa độ cứng sẽ sai ngay.

Cơ chế bên dưới: PDF Object Tree

Để hiểu tại sao `pdfplumber` thất bại, ta phải nhìn vào **cấu trúc dữ liệu** của PDF.

PDF không phải là văn bản phẳng, nó là một cây đối tượng (Object Tree):

File Digital (Gốc)

Cấu trúc cây phong phú:

- `/Page`
 - `/Resources`
 - `/Contents`
 - `TextObject` (Chứa tọa độ & chữ)
 - `PathObject` (Đường kẻ bảng)

`pdfplumber` duyệt cây này để lấy dữ liệu.

File Scan (Ảnh)

Cấu trúc cây không có các Text Object:

- `/Page`
 - `/Resources`
 - `/Contents`
 - `XObject (Image)`
 - *(Không có TextObject nào!)*

`pdfplumber` tìm `TextObject` → Không thấy → **Rỗng.**

3. Giải pháp: Computer Vision (AI)

Khi cấu trúc cây (Object Tree) không chứa dữ liệu văn bản, ta cần một mô hình bỏ qua cấu trúc cây và **"nhìn"** vào điểm ảnh (Pixels) như con người.

Pipeline hiện đại (Ví dụ: LayoutLM, Dolphin, Azure DI):

1. **Vision Encoder:** Chuyển ảnh trang giấy thành các điểm ảnh (pixels).
2. **OCR:** Phát hiện hình dáng các con chữ (kể cả khi bị nghiêng hay mờ).
3. **Layout Analysis:** Hiểu được "Khối này là Bảng" và "Khối kia là Tiêu đề".

*Nó không quan tâm code bên dưới là gì. Nó chỉ quan tâm tài liệu **trông** như thế nào.*

DEMO TIME (PHẦN 2)

Cách tiếp cận "AI / Vision"

Kịch bản: Chúng ta sẽ thử đọc LẠI file **Hóa đơn Scan** (mà lúc nãy bị lỗi) bằng công nghệ Computer Vision.

(Chuyển sang màn hình Streamlit để xem AI vẽ khung hình)

Kết quả Demo (Recap)

Nếu demo chạy đúng, các bạn sẽ thấy sự khác biệt rõ rệt của 2 phương pháp này:

9				
10				
HOANGKIM JSC				
Cộng tiền hàng :				
Thuế suất GTGT :.....%			Tiền thuế GTGT:	
Tổng số tiền thanh toán :				
Số tiền viết bằng chữ :.....				
Người mua hàng			Người bán hàng	
<div>Mẫu: Ký bởi: CÔNG TY CỔ PHẦN VẬT TƯ THIẾT BỊ HOÀNG KIM Ký ngày:</div>				

Visualizing Bounding Boxes (How pdfplumber sees text)

Failed (0.0762s)

Analysis: Returns Empty. No selectable text layer found.

Conclusion: This is a scanned document/image. Traditional parsers cannot read it.



HÓA ĐƠN
GIÁ TRỊ GIA TĂNG
Ngày tháng năm

Mẫu số : 01GKT0/001
Ký hiệu : HK/18E
Số : 0000000

CÔNG TY CỔ PHẦN VẬT TƯ THIẾT BỊ HOÀNG KIM
Mã số thuế : 0108397618
Địa chỉ : Số nhà 18, ngõ 114, đường Thanh Bình, Phường Mộ Lao, Quận Hà Đông, Thành phố Hà Nội, Việt Nam
Số tài khoản : 19133030911017 - Ngân hàng Techcombank - CN Hà Tây



Tên người mua hàng :
Tên đơn vị :
Mã số thuế :
Địa chỉ :
Hình thức thanh toán :
Số tài khoản :

STT	Tên hàng hóa, dịch vụ	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
1	2	3	4	5	6=4×5
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Cơ chế bên dưới: AI nhìn thấy gì?

Thay vì tìm tọa độ (x, y) trong metadata, AI xử lý **Mảng các điểm ảnh (Pixel Matrix)**.

Quy trình xử lý mà Demo vừa chạy:

1. **Preprocessing:** Chuyển PDF thành Ảnh (Image) → Khử nhiễu.
2. **Text Detection:** Khoanh vùng "Ở đâu có chữ?" (Chính là các **ô màu xanh** trong demo).
3. **Text Recognition:** Cắt ô xanh đó ra và hỏi "Đây là chữ gì?".
4. **Layout Analysis:** Ghép các chữ lại thành dòng và bảng.

Sức mạnh: AI "đoán" chữ dựa trên hình dáng → Đọc được file scan, chữ viết tay.

4. So sánh & Chiến lược

Khi nào nên dùng cái nào?

Tính năng	Truyền thống (<code>pdfplumber</code>)	AI Hiện đại (OCR/Vision)
Phương pháp	Đọc Metadata của PDF	Đọc Pixels
Tốc độ	Siêu nhanh (ms)	Chậm hơn (giây)
Chi phí	Miễn phí / Tốn ít CPU	Cao (Cần GPU / API)
File Scan	Bó tay	Xuất sắc
Độ chính xác	100% (nếu là file gốc)	~95-99% (Xác suất)

Kết luận: Dùng phương pháp **Truyền thống** cho file digital số lượng lớn. Dùng **AI** làm phương án dự phòng (fallback) cho file xấu/scan.

5. Beyond the Basics: Enterprise Tools

Trong thực tế Production, chúng ta thay thế script thủ công bằng các công cụ chuyên dụng.

Microsoft Dolphin / LayoutLM

"The Brain" — Bộ não

- **Loại:** Multimodal AI
- **Khả năng:** Không chỉ OCR (đọc chữ) mà còn hiểu *ngữ nghĩa & bố cục*
- **Ví dụ:** Tự biết cụm chữ in đậm ở góc trên là **"Tên Công Ty"** chứ không chỉ là text rời rạc.

Unstructured.io

"The Manager" — Bộ quản lý luồng dữ liệu

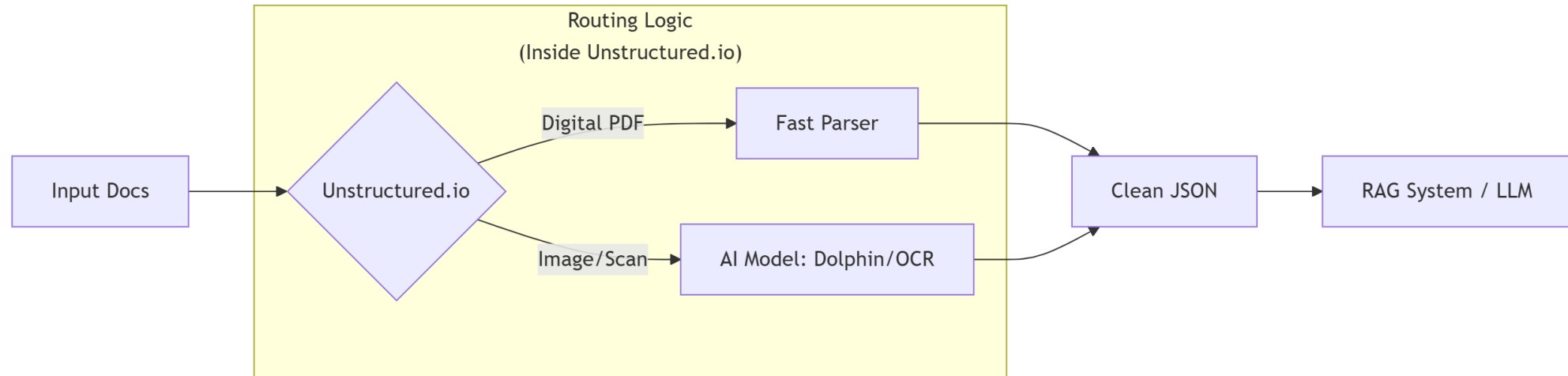
- **Loại:** ETL / Pre-processing Engine
- **Khả năng:** Tự động **routing thông minh**
 - File đẹp (Digital) → Fast Parser (chi phí thấp)
 - Scan mờ → Gọi AI Model (chi phí cao)

Kết luận:

Dolphin giúp xử lý document — Unstructured giúp tự động hóa việc chọn công cụ.

6. The Ultimate Goal: Automated Pipeline

Kết hợp tất cả lại, đây là kiến trúc chuẩn cho một hệ thống **RAG** hoặc **Data Entry** tự động:



Thông điệp: "Unstructured.io dùng để quản lý luồng xử lý file, còn pdfplumber hay Dolphin là các "engine" xử lý ở mỗi luồng."

Cảm ơn mọi người!

Công cụ sử dụng:

- **Frontend:** Streamlit
- **Thư viện:** pdfplumber
- **Concept:** OCR & Vision Transformers