Structured data extraction from unstructured content using LLM schemas

axe finance

Focus · Expertise · Value

Presented by

Malek Zitouni

Overview

1 Introduction & Context

- KYC documents for credit applications come in countless formats, scanned PDFs, images, tables, handwritten notes, making **data extraction** messy and inconsistent.

- Traditional systems fail when **layouts or labels change**

- ***Our solution*** : An **LLM-powered adaptive extraction system** that takes:
  - An unstructured document
  - A dynamic schema of required fields

It finds the right data, maps it to the schema, and outputs **structured results** , no manual reconfiguration, scalable for evolving formats.

Invoice no: 51109338

Date of issue: 04/13/2013

Seller:

Andrews, Kirby and Valdez
58861 Gonzalez Prairie
Lake Daniellefurt, IN 57228

Tax Id: 945-82-2137
IBAN: GB75MCRL06841367619257

Client:

Becker Ltd
8012 Stewart Summit Apt. 455
North Douglas, AZ 95355

Tax Id: 942-80-0517

ITEMS

No.	Description	Qty	UM	Net price	Net worth	VAT [%]	Gross worth
1.	CLEARANCE! Fast Dell Desktop Computer PC DUAL CORE WINDOWS 10 4/8/16GB RAM	3,00	each	209,00	627,00	10%	689,70
2.	HP T520 Thin Client Computer AMD GX-212JC 1.2GHz 4GB RAM TESTED !!READ BELOW!!	5,00	each	37,75	188,75	10%	207,63
3.	gaming pc desktop computer	1,00	each	400,00	400,00	10%	440,00
4.	12-Core Gaming Computer Desktop PC Tower Affordable GAMING PC 8GB AMD Vega RGB	3,00	each	464,89	1 394,67	10%	1 534,14
5.	Custom Build Dell Optiplex 9020 MT i5-4570 3.20GHz Desktop Computer PC	5,00	each	221,99	1 109,95	10%	1 220,95
6.	Dell Optiplex 990 MT Computer PC Quad Core i7 3.4GHz 16GB 2TB HD Windows 10 Pro	4,00	each	269,95	1 079,80	10%	1 187,78
7.	Dell Core 2 Duo Desktop Computer | Windows XP Pro | 4GB | 500GB	5,00	each	168,00	840,00	10%	924,00

SUMMARY

VAT [%]	Net worth	VAT	Gross worth
10%	5 640,17	564,02	6 204,19
Total	**$ 5 640,17**	**$ 564,02**	**$ 6 204,19**

Exemple de nom d'entreprise
123 Impasse Eugène Boudin
83130 La Garde
Siret : 98141753800015 **Code NAF :** 4932Z
Urssaf :
Convention : 0016 - Convention collective nationale des transports routiers et activités auxiliaires du transport

BULLETIN DE PAIE

Période du : 01/05/2025 **au :** 31/05/2025
Paiement le : 31/05/2025

Matricule : 001
N° Sécurité Sociale : 172023155521988
Entrée le : 01/11/2023
Emploi : Président
Qualification : Non Cadre
Coefficient : 225
Plafond Sécurité Sociale : 3 925,00 €
Contrat : CDI

IMAN MAHROUG
321 Impasse Eugène Boudin,
83130 La Garde

Rubriques	Base	Taux Salarial	Cot. Salariales	Taux Patronal	Cot. Patronales
SALAIRE DE BASE	5 200,00 €	34,28			
SALAIRE BRUT	5 200,00 €				
SANTÉ					
Sécurité sociale	5 200,00 €			7,00	364,00 €
ACCIDENTS DU TRAVAIL - MALADIES PROFESSIONNELLES	5 200,00 €			2,23	115,96 €
RETRAITE					
Assurance Vieillesse déplafonnée	5 200,00 €	0,40	20,80 €	2,02	105,04 €
Assurance Vieillesse plafonnée	3 925,00 €	6,90	270,83 €	8,55	335,59 €
Retraite complémentaire tranche: AGIRC-ARRCO	5 200,00 €	3,15	163,80 €	4,72	245,44 €
CEG tranche 1	5 200,00 €	0,86	44,72 €	1,29	67,08 €
FAMILLE-SÉCURITÉ SOCIALE	5 200,00 €				135,42 €
ASSURANCE CHÔMAGE					
Assurance chômage tranche A	5 200,00 €			4,05	210,60 €
Assurance chômage tranche AGS (FNGS)	5 200,00 €			0,25	13,00 €
CSG déductible	5 109,00 €	6,80	347,41 €		
AUTRES CONTRIBUTIONS DUES PAR L'EMPLOYEUR	5 200,00 €				85,80 €
TOTAL DES RETENUES			**847,56 €**		**1 677,93 €**
CSG/CRDS imposable à l'impôt sur le revenu	5 109,00 €	2,90	148,16 €		

MONTANT NET SOCIAL	4 352,44 €
NET À PAYER AVANT IMPOT SUR LE REVENU	**4 204,28 €**
Dont évolution de la rémunération liée à la suppression des cotisations chômage et maladie	76,95 €

Heures période	151,67	Cumul bases	26 000,00 €	Total cot. patronales	1 677,93 €
Cumul heures	758,35	Cumul bruts	26 000,00 €	Total des retenues	2 525,49 €
Cumul heures sup.	0,00	Cumul imposable	21 762,20 €	Coût global periode	6 877,93 €

Impot sur le revenu	Base	Taux Neutre	Montant	Cumul annuel
Montant net imposable			4 352,44 €	21 762,20 €
Impôt sur le revenu prélevé à la source	4 352,44 €	17,10 %	744,27 €	3 721,34 €
Montant net des heures compl/suppl exonérées	0,00 €			

Congés Payés			
En cours	0,00	**NET À PAYER AU SALARIÉ**	**3 460,01 €**
Acquis	0,00		
Pris	0,00		
Solde	0,00		

2 Project Overview

Our system transforms complex, **unstructured KYC documents (PDFs/images)** into clean, **structured data**, guided by dynamic schemas.

Key Workflow

1. **Document Layout Analysis (DLA)** : Detects structural elements (text, tables, headers, lists, forms) using RT-DETR-based models trained on 17 layout labels.

2. Table Structure Recognition (TSR) : Identifies rows, columns, cells using Microsoft's Table Transformer.

3. OCR Extraction : Extracts text from detected regions (tables, paragraphs, etc.).

4. Schema-Based Mapping with LLMs : Matches extracted text to user defined schema fields.

Phase 1: OCR Extraction Pipeline

Current Pipeline:

1. Layout Analysis → Table Structure Recognition → OCR Extraction

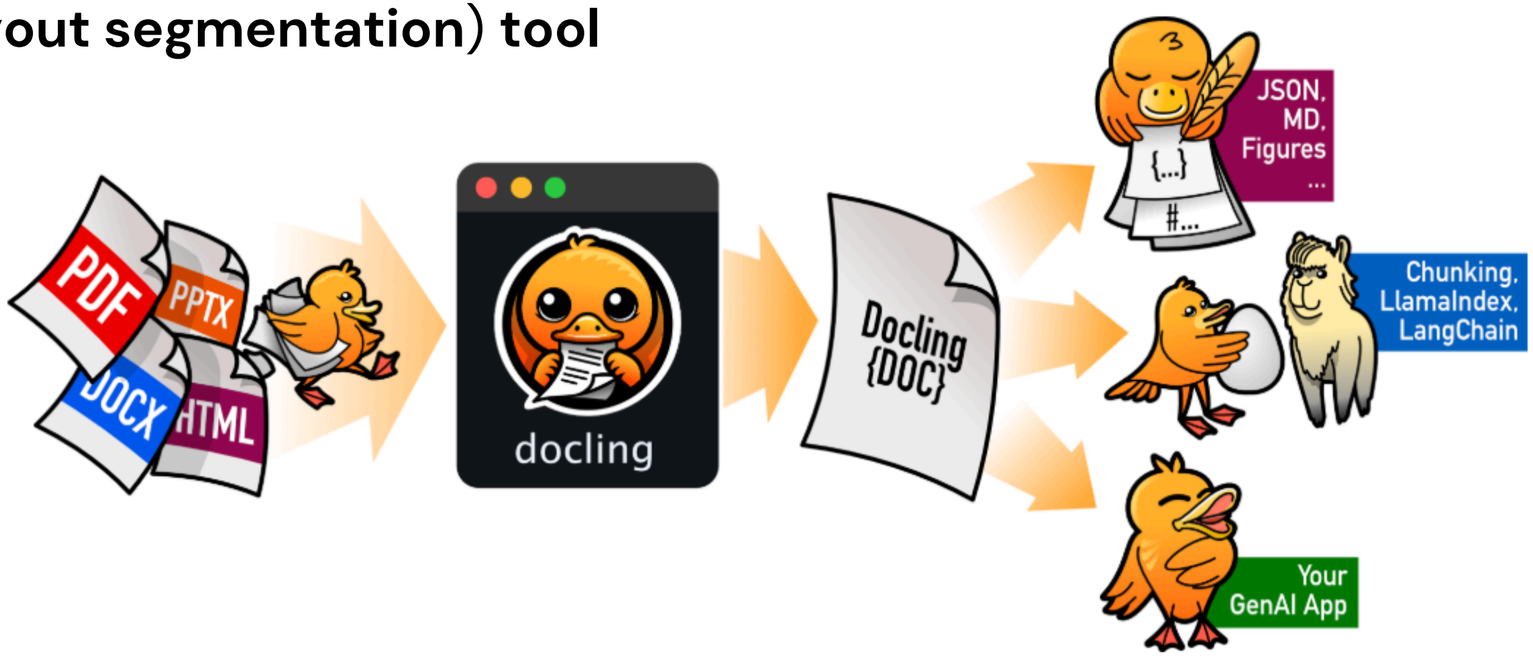
3 Technical Approach

Document Layout Analysis (DLA)

- Our system begins with Document Layout Analysis (DLA) , segmenting each KYC document into meaningful **regions** before extraction.

Docling

It’s an open-source Python framework designed to turn virtually any document format into a unified, structured representation suitable for downstream AI tasks. it’s a also document layout analysis (or **layout segmentation**) **tool**



3 Technical Approach

Document Layout Analysis (DLA)

DLA Models Used:

* ds4sd/docling-layout-old
* ds4sd/docling-layout-heron
* ds4sd/docling-layout-heron-101
* ds4sd/docling-layout-egret-medium
* ds4sd/docling-layout-egret-large

Purpose:

- Identify text blocks, tables, headers, lists, and forms ....
- Provide **structured layout metadata** for downstream OCR and LLM-based mapping
- Ensure **layout-agnostic extraction**, even for complex or inconsistent document designs

Outcomes:

A clean, machine-readable document structure , ready for precise schema-based data mapping

Labels

```
0: "Caption",
1: "Footnote",
2: "Formula",
3: "List-item",
4: "Page-footer",
5: "Page-header",
6: "Picture",
7: "Section-header",
8: "Table",
9: "Text",
10: "Title",
11: "Document Index",
12: "Code",
13: "Checkbox-Selected",
14: "Checkbox-Unselected",
15: "Form",
16: "Key-Value Region",
```

```
{
  "l": 131.99163818359375,
  "t": 827.824951171875,
  "r": 1520.9412841796875,
  "b": 1663.9490966796875,
  "label": "Table",
  "confidence": 0.9906219244003296
},
```


(left x-coordinate of the bounding box)
(top y-coordinate of the bounding box)
(right x-coordinate of the bounding box
(bottom y-coordinate of the bounding box)


```
{
  "l": 825.956787109375,
  "t": 444.9659729003906,
  "r": 947.0271606445312,
  "b": 473.4598693847656,
  "label": "Section-header",
  "confidence": 0.907884955406189
},
```

Table Structure Recognition (TSR)

After Document Layout Analysis, tables are passed to a Table Structure Recognition (TSR) module to understand their internal organization.

Tool Used:

- The **TableFormer Model** has achieved state-of-the-art table structure identification. To use this model, we provided **an image** of a table as input, along with **the predicted table regions from the Layout Model.**

Purpose:

- Detect table boundaries
- Identify rows, columns, and cell coordinates
- Preserve the logical reading order
- Output cell structures for precise OCR text extraction

Why It Matters:

- Supports multi-page and irregular table layouts

Outcome:

A fully structured table grid with coordinates and metadata ready for cell-level OCR and schema mapping

```
{
  "label": "table row",
  "score": 0.094,
  "box": [
    34.8,
    355.1,
    1363.8,
    465.3
  ],
  "table_bbox": {
    "l": 131.99163818359375,
    "t": 827.824951171875,
    "r": 1520.9412841796875,
    "b": 1663.9490966796875,
    "label": "Table",
    "confidence": 0.9906219244003296
  }
},
```

Document Block OCR Extraction

Main Points:

1. **Input**
   - Document image (invoice, receipt, form)
   - Layout JSON: block coordinates + labels from prior layout analysis
   - Optional: target labels (e.g., only "Table", "Section-header")
2. **Process**
   - Load NanonetsExtractor OCR model from local cache
   - For each block:
     - Identify block type
     - Pick a custom prompt template (table, text, title, etc.)
     - Crop image region using bounding box
     - Send cropped image + prompt to OCR extractor
3. **Output**
   - For each block:
     - Block ID, label, confidence score, bounding box, extracted text
   - Save in JSON (structured data) and TXT (human-readable)

Nanonets-OCR-s

- It is a powerful, state-of-the-art image-to-markdown OCR model that goes far beyond traditional text extraction. It transforms **documents into structured markdown** with intelligent content recognition and semantic tagging, making it ideal for downstream processing by Large Language Models (LLMs).

Nanonets-OCR-s – Key Capabilities

1. LaTeX Equation Recognition
2. Intelligent Image Description: Generates structured <img> tags describing embedded images (logos, charts, graphs, etc.), including their content, style, and context for LLM-friendly processing.
3. Signature Detection & Isolation
4. Watermark Extraction
5. Complex Table Extraction: Accurately captures intricate table layouts and converts them into both Markdown and HTML formats.

```
{
  "block_id": 2,
  "label": "Text",
  "confidence": 0.94014972448349,
  "bounding_box": [
    832.8427734375,
    506.4781799316406,
    1263.6275634765625,
    610.233642578125
  ],
  "text": "Becker Ltd\n8012 Stewart Summit Apt. 455\nNorth Douglas, AZ 95355"
```

```
  "block_id": 9,
  "label": "Section-header",
  "confidence": 0.8872629404067993,
  "bounding_box": [
    132.4087677001953,
    1715.204345703125,
    327.2047119140625,
    1743.799560546875
  ],
  "text": "SUMMARY"
},
```

```
=== BLOCK 1 ===
Type: Table
Confidence: 0.99
Position: [131.99163818359375, 827.824951171875, 1520.9412841796875, 1663.9490966796875]

EXTRACTED TEXT:
+------------+----------+---------+---------+---------+---------+---------+
| No.        | Description | Qty    | UM      | Net price | Net worth | VAT [%] | Gross worth |
|            |           |         |         |           |           |         |             |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 1.         | CLEARANCE! Fast Dell Desktop Computer PC DUAL CORE WINDOWS 10 4/8/16GB RAM | 3,00    | each     | 209,00    | 627,00    | 10%      | 689,70      |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 2.         | HP T520 Thin Client Computer AMD GX-212JC 1.2GHz 4GB RAM TESTED!!READ BELOW!! | 5,00    | each     | 37,75     | 188,75    | 10%      | 207,63      |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 3.         | gaming pc desktop computer | 1,00    | each     | 400,00    | 400,00    | 10%      | 440,00      |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 4.         | 12-Core Gaming Computer Desktop PC Tower Affordable GAMING PC 8GB AMD Vega RGB | 3,00    | each     | 464,89    | 1 394,67  | 10%      | 1 534,14    |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 5.         | Custom Build Dell Optiplex 9020 MT i5-4570 3.20GHz Desktop Computer PC | 5,00    | each     | 221,99    | 1 109,95  | 10%      | 1 220,95    |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 6.         | Dell Optiplex 990 MT Computer PC Quad Core i7 3.4GHz 16GB 2TB HD Windows 10 Pro | 4,00    | each     | 269,95    | 1 079,80  | 10%      | 1 187,78    |
+------------+----------+---------+---------+---------+---------+---------+-------------+
| 7.         | Dell Core 2 Duo Desktop Computer | Windows XP Pro | 4GB | 500GB | 5,00    | each     | 168,00    | 840,00    | 10%      | 924,00      |
+------------+----------+---------+---------+---------+---------+---------+-------------+

================================================

=== BLOCK 2 ===
Type: Table
Confidence: 0.97
Position: [131.67897033691406, 1779.6932373046875, 1520.6849365234375, 1960.77197265625]

EXTRACTED TEXT:
| VAT [%] | Net worth | VAT | Gross worth |
|---------|-----------|-----|-------------|
|         | 10%       |     |             |
| Total   | $ 5 640,17 | $ 564,02 | $ 6 204,19 |

|         | VAT | Gross worth |
|---------|-----|-------------|
|         | 564,02 | 6 204,19    |
```

```
=== BLOCK 1 ===
Type: Section-header
Confidence: 0.92
Position: [133.9689178466797, 760.8466186523438, 245.8738555908203, 789.2060546875]

EXTRACTED TEXT:
ITEMS

================================================

=== BLOCK 2 ===
Type: Section-header
Confidence: 0.
Position: [825.956787109375, 444.9659729003906, 947.0271606445312, 473.4598693847656]

EXTRACTED TEXT:
Client:

================================================

=== BLOCK 3 ===
Type: Section-header
Confidence: 0.89
Position: [132.4087677001953, 1715.204345703125, 327.2047119140625, 1743.799560546875]

EXTRACTED TEXT:
SUMMARY

================================================

=== BLOCK 4 ===
Type: Section-header
Confidence: 0.89
Position: [132.9769744873047, 444.42156982421875, 252.46241760253906, 473.6143493652344]

EXTRACTED TEXT:
Seller:
```

Phase 2: Schema-Based Mapping Pipeline

Detailed Logic Explanation

Purpose

Transform unstructured document text (from OCR) into structured data by mapping field values to predefined schema fields using AI models.

Let:

- M = number of schema fields
- N = number of OCR text blocks
- Grid size = M×N

Step 1 : Schema Enhancement

For each f_i where $i \in [1, M]$:

- TinyLlama → generates:
  - $Syn(f_i)$ = synonyms
  - $Kw(f_i)$ = context keywords
  - $Inst(f_i)$ = extraction instructions

  → Enriched field definition $EFD(f_i)$

Step 2 : Field Extraction Loop

For each b_j where $j \in [1, N]$:

For each f_i where $i \in [1, M]$:

1. **Regex**: match patterns like `"f_i: v"` or `"f_i = v"` using $Syn(f_i)$
2. If fail → **LLM (Mistral)** extraction with context prompt
3. Store candidate:

$$C_{i,j} = \{value, blockID = j, method, confidence\}$$

-The goal of using regex is to **quickly and precisely extract structured "field–value" pairs from raw text before trying more complex (and slower) methods like an LLM**.

- **Regex(Regular expressions) Pattern Matching**

```
Patterns Used:
1. Field: Value (field_name: extracted_value)
2. Field = Value (field_name = extracted_value)
3. Field Value (field_name extracted_value)
4. Field (Value) (field_name(extracted_value))

Confidence Scoring:
- Pattern 1 (Colon): 0.9 confidence
- Pattern 2 (Space): 0.8 confidence
- Pattern 3 (Parentheses): 0.7 confidence
- Data-type specific: 0.6-0.7 confidence
```

Pattern	Example	Regex Style
1 — Colon	`Field: Value`	`^(\w+):\s*(.+)$`
2 — Equals	`Field = Value`	`^(\w+)\s*=\s*(.+)$`
3 — Space	`Field Value`	`^(\w+)\s+(.+)$`
4 — Parentheses	`Field (Value)`	`^(\w+)\s*\((.+)\)$`

```
            # Generate field metadata
            prompt = f"""
You are a field analysis expert. Analyze the field definition and generate:
1. Up to 5 common synonyms that might appear in documents
2. Specific extraction instructions based on field characteristics
3. Context keywords that might appear near the value in text

FIELD: {field_name}
DESCRIPTION: {field_def.description}
DATA TYPE: {field_def.data_type}

RESPONSE FORMAT:
{{
  "synonyms": ["synonym1", "synonym2", ...],
  "extraction_instructions": "Detailed instructions for value extraction",
  "context_keywords": ["keyword1", "keyword2", ...]
}}

Respond with ONLY the JSON object, no explanations.
"""
```

```
        prompt = f"""
You are an expert document field extractor. Analyze the text block to identify field values using their descriptions and synonyms.

FIELD DEFINITIONS:
{chr(10).join(field_descriptions)}

TEXT BLOCK (ID: {block_id}):
{block_text}

EXTRACTION RULES:
1. Match fields using any known synonyms from the definitions
2. For numbers: extract digits only (e.g., "Age: 25 years" → "25")
3. For dates: preserve original format or convert to YYYY-MM-DD
4. For alphanumeric codes: preserve exact formatting (e.g., "4932Z")
5. If a field appears multiple times, extract each occurrence
6. Return JSON with ALL field names from the schema
7. For missing fields: set "exists" to false and "extracted_value" to empty string
8. Ignore punctuation differences between field names and text (e.g., "Field:" matches "Field")

REQUIRED RESPONSE FORMAT:
{{
  "field1": {{"exists": boolean, "extracted_value": "string"}},
  "field2": {{"exists": boolean, "extracted_value": "string"}},
  ... (ALL FIELDS MUST BE INCLUDED)
}}

Respond with ONLY the JSON object, no explanations.
"""
```

Conflict Resolution System

Step 1: Conflict Detection

Validation Logic:
1. Single candidate → No conflict
2. Multiple identical values → No conflict
3. High similarity values (>80%) → No conflict
4. Multiple different values → Validate with TinyLlama

TinyLlama Validation:
- Analyzes if differences are meaningful
- Considers formatting variations
- Returns CONFLICT or NO_CONFLICT decision

Step 2: Conflict Resolution

Mistral Resolution Process:
1. Present all candidate options with context
2. Apply selection criteria:
   - Accuracy to field description
   - Extraction method preference (regex > llm)
   - Higher extraction confidence
   - Higher OCR confidence
   - More complete context
3. Return selected value with reasoning
4. Validate selection against candidate list
5. Fallback to scoring system if invalid selection

Conflict Resolution System

Step 3: Scoring Fallback System

```
Scoring Algorithm:
score = 0
if extraction_method == "regex": score += 3
elif extraction_method == "llm": score += 2

score += extraction_confidence * 2
score += ocr_confidence * 1

best_candidate = max(candidates, key=lambda x: x.score)
```

- ***Step 2 — Candidate Resolution***

For each f_i:

- If $|Cand(f_i)| = 1$ → select directly
- If all v_k identical → select highest confidence
- If $|UniqueVals(f_i)| > 1$:
  - TinyLlama → validate if conflict is real (≠ formatting diff)
  - If real → Mistral chooses best v^* based on:
    - Field description match
    - Data type consistency
    - Context relevance

```
        prompt = f"""
You are a conflict validator. Analyze if these candidate values represent a real conflict.

FIELD: {field_name}
CANDIDATE VALUES:
{chr(10).join(f"- {val}" for val in candidate_values)}

VALIDATION RULES:
1. Check if values are truly different and conflicting
2. Consider if they might be equivalent representations
3. Ignore minor formatting differences
4. Return "CALLBACK_NEEDED" only for genuine conflicts

RESPONSE FORMAT:
- Return "CALLBACK_NEEDED" if values are conflicting
- Return "NO_CALLBACK" if values are equivalent
- Respond with ONLY the decision keyword

Decision:"""
```

Model 1: Mistral-7B-Instruct-v0.2(Primary Extraction Engine)

Role: Main field extraction and conflict resolution
Capabilities:
- Complex reasoning and pattern recognition
- Multi-field extraction from text blocks
- Contextual understanding
- Conflict resolution decision making

Parameters:
- Model Type: Causal Language Model
- Max Tokens: 2048 for extraction tasks
- Temperature: 0.0 (deterministic output)
- Format: Chat template with structured prompts

Model 2: TinyLlama-1.1B (Validation & Analysis)

Role: Field analysis and validation support
Capabilities:
- Fast synonym generation
- Field metadata enhancement
- Conflict validation
- Lightweight processing

Parameters:
- Model Type: Lightweight Language Model
- Max Tokens: 500 for analysis tasks
- Temperature: 0.0 (consistent results)
- Format: System/User/Assistant template

```
{
 "address_client": {
    "description": "Client's street address","data_type": "string"
},
 "address_seller": {
    "description": "seller's street address","data_type": "string"
}
}
```


Seller:

Andrews, Kirby and Valdez
58861 Gonzalez Prairie
Lake Daniellefurt, IN 57228

Tax Id: 945-82-2137
IBAN: GB75MCRL06841367619257

Client:

Becker Ltd
8012 Stewart Summit Apt. 455
North Douglas, AZ 95355

Tax Id: 942-80-0517

results_mapping > {} mapping_values_only.json > ...


```
{
  "address_client": "58861 Gonzalez Prairie, Lake Daniellefurt, IN 57228",
  "address_seller": "8012 Stewart Summit Apt. 455, North Douglas, AZ 95355"
}
```

sults_mapping > {} mapping_final_results.json > ...


```json
{
  "structured_data": {
    "address_client": "58861 Gonzalez Prairie, Lake Daniellefurt, IN 57228",
    "address_seller": "8012 Stewart Summit Apt. 455, North Douglas, AZ 95355"
  },
  "source_traceability": {
    "address_client": "3",
    "address_seller": "2"
  },
  "dual_llm_usage_summary": {
    "mistral_extraction_calls": 40,
    "mistral_callback_calls": 0,
    "total_mistral_calls": 40,
    "tinyllama_validation_calls": 2,
    "architecture": "TinyLlama validator with Mistral callbacks"
  }
}
```

```
{
  "block_id": 2,
  "label": "Text",
  "confidence": 0.94014972448349,
  "bounding_box": [
    832.8427734375,
    506.4781799316406,
    1263.6275634765625,
    610.2336425781 25
  ],
  "text": "Becker Ltd\n8012 Stewart Summit Apt. 455\nNorth Douglas, AZ 95355"
},
```

```
{
  "block_id": 3,
  "label": "Text",
  "confidence": 0.9339390397071838,
  "bounding_box": [
    140.08267211914062,
    506.8832092285156,
    524.1350708007812,
    605.605712890625
  ],
  "text": "Andrews, Kirby and Valdez\n58861 Gonzalez Prairie\nLake Daniellefurt, IN 57228"
},
```

```
"date_of_issue": {
  "description": "Date the invoice was issued in MM/DD/YYYY format.",    "data_type": "string"
},
```

Invoice no: 51109338

Date of issue: 04/13/2013

```
{
  "date_of_issue": "04/13/2013"
}
```

```
{
  "date_of_issue": "14"
}
```

```
  "invoice_no": {
    "description": "Unique invoice identification number",
    "data_type": "string"
  },
  "tax_id": {
        "description": "Seller's tax identification number",
        "data_type": "string"
      },
  "date_of_issue": {
    "description": "Date the invoice was issued in MM/DD/YYYY format",
    "data_type": "date"
  },
  "iban": {
        "description": "Seller's International Bank Account Number",
        "data_type": "string"
      },
  "address_client" : {
        "description": "Client's street address",
        "data_type": "string"
      },

  "address_seller" : {
        "description": "Client's street address",
        "data_type": "string"
      },

  "net_worth": {
          "description": "Total price for the quantity excluding tax",
          "data_type": "string"
        },
  "vat_percent": {
          "description": "Value Added Tax rate applied to the item",
          "data_type": "string"
        },
  "gross_worth": {
          "description": "Total price including tax",
          "data_type": "string"
        },

  "gross_worth_total": {
        "description": "Total gross worth including tax",
        "data_type": "string"
      }
```

Invoice no: 51109338

Date of issue: 04/13/2013

Seller:

Andrews, Kirby and Valdez
58861 Gonzalez Prairie
Lake Daniellefurt, IN 57228

Tax Id: 945-82-2137
IBAN: GB75MCRL06841367619257

Client:

Becker Ltd
8012 Stewart Summit Apt. 455
North Douglas, AZ 95355

Tax Id: 942-80-0517

VAT [%]	Net worth	VAT	Gross worth
10%	5 640,17	564,02	6 204,19
Total	**$ 5 640,17**	**$ 564,02**	**$ 6 204,19**

VAT [%]
10%
10%
10%
10%
10%
10%
10%

results_mapping > {} mapping_batch_values.json > ...


```json
{
  "invoice_no": "51109338",
  "tax_id": "DE123456789",
  "date_of_issue": "04/13/2013",
  "iban": "GB75MCRL06841367619257",
  "address_client": "Becker Ltd\n8012 Stewart Summit Apt. 455\nNorth Douglas, AZ 95355",
  "address_seller": "Seller Street 2, 2020 City, Germany",
  "net_worth": "5640.17",
  "vat_percent": "10%",
  "gross_worth": "6204.19",
  "gross_worth_total": "6204.19"
}
```

Tools & Technologies Used

1. Deep Learning Framework

- PyTorch: Core ML framework for model loading and inference
- GPU Acceleration: CUDA support for faster processing
- 4-bit Quantization: Memory-efficient model loading

2. Natural Language Processing

- Transformers Library: HuggingFace transformers for model handling
- Text Generation Pipeline: Structured text generation
- Chat Templates: Proper prompt formatting for each model

3. Data Validation & Processing

-Pydantic: Data validation and serialization

-JSON Schema: Structured data definitions

-Regular Expressions: Pattern matching and text cleaning

4. Programming Libraries

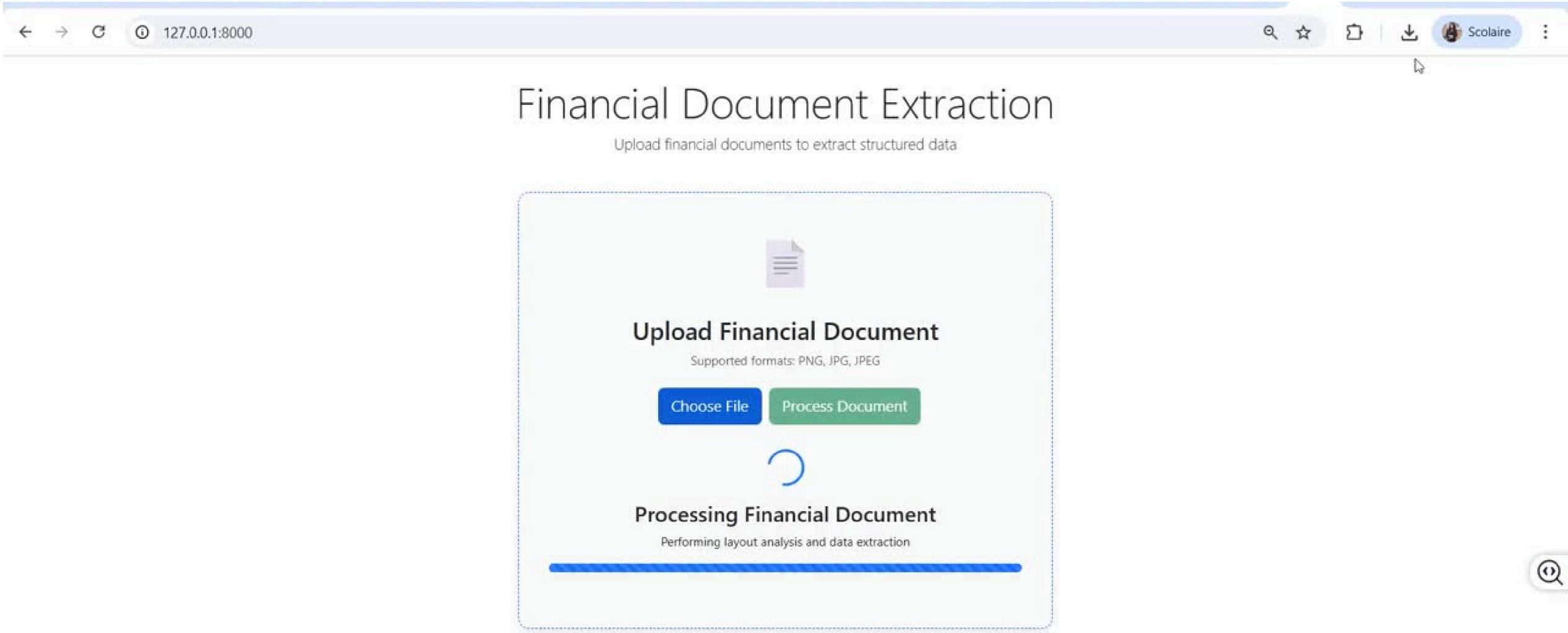
-Python 3.8+: Main programming language

-Collections.Counter: Frequency analysis for conflict resolution

-difflib.SequenceMatcher: Text similarity calculations

-Logging: Comprehensive system monitoring

4. Results & Impact



5. Future Outlook

Phase 1: Core Enhancement

- Implement spatial relationship mapping
- Add confidence-based filtering
- Create pattern-matching database

Phase 2: LLM Optimization

- Implement batch processing for LLM calls
- Enhance conflict resolution logic
- Create a validation pipeline

Phase 3: Advanced Features

- Add schema learning capabilities
- Implement multi-modal understanding
- Create comprehensive testing framework

Thank you