**Handling Multiple Deeds on a Single Deed Book Page**

When training Google Document AI to extract data from pages containing multiple deeds, you need a systematic approach that ensures accurate identification and separation of each instrument. Here's a comprehensive strategy for addressing this challenge:

**Document Pre-processing Techniques**

1. **Implement visual segmentation algorithms** that identify natural boundaries between deeds based on:
   * Distinct header sections or title blocks
   * Significant whitespace or ruled lines between instruments
   * Changes in formatting or indentation patterns
2. **Apply document layout analysis** as a preliminary step before extraction:
   * Configure the processor to recognize multi-column layouts
   * Train the model to identify typical deed structural patterns
   * Use spatial relationship analysis to distinguish separate instruments

**Training Data Preparation**

1. **Create specialized annotations** that explicitly mark:
   * The beginning and end boundaries of each deed
   * The relationship between a deed and its component parts
   * Areas where deeds adjoin or overlap
2. **Develop a hierarchical labeling schema** that includes:
   * Page-level entities
   * Individual deed-level entities
   * Field-level entities within each deed
3. **Overrepresent multiple-deed pages** in your training data to ensure the model develops robust capabilities for this specific challenge.

**Model Configuration Strategies**

1. **Design a two-stage extraction process**:
   * First stage: Identify and segment individual deeds on the page
   * Second stage: Extract specific data fields from each identified deed
2. **Configure entity relationships** in Document AI Workbench:
   * Define a "Deed" entity type that can occur multiple times per page
   * Create parent-child relationships between the page and each deed
   * Establish consistent identifiers to maintain connections between segmented deeds
3. **Implement confidence thresholds** specifically calibrated for deed separation decisions.

**Post-processing Validation**

1. **Apply structural validation rules** to verify each extracted deed contains:
   * Proper identifying information (instrument type, recording information)
   * Essential deed components (grantor, grantee, legal description)
   * Internal consistency in referenced data
2. **Develop deed completeness checks** that flag potential segmentation errors:
   * Deeds missing critical components
   * Unusually short or long extractions
   * Inconsistent formatting within a supposed single deed

**Implementation Recommendations**

1. **Create a custom processor** specifically optimized for multiple-deed scenarios rather than using general-purpose document extractors.
2. **Incorporate metadata extraction** that captures:
   * The total number of deeds on the page
   * The sequential position of each deed
   * Cross-references between deeds when applicable
3. **Establish a human review workflow** for cases where:
   * Confidence scores fall below thresholds
   * Structural validation rules flag potential issues
   * Typical deed components appear to be missing

This multi-faceted approach addresses the technical challenges while ensuring the practical needs of land records extraction are met. By implementing these strategies, your Document AI model will more effectively handle the complexities of multiple deeds on a single page, resulting in higher accuracy extraction results.

**Training Google Document AI for Recorded Document Data Extraction**

**Prerequisites and Planning**

Before initiating the training process, it is essential to establish a clear foundation:

1. Define your extraction goals with precision, identifying which specific data fields from recorded documents (e.g., parcel numbers, grantors/grantees, legal descriptions) are priorities.
2. Secure access to Google Cloud Platform with Document AI API enabled and appropriate permissions configured.
3. Develop a comprehensive taxonomy of data fields that aligns with your extraction requirements.
4. Allocate sufficient resources for the annotation process, which is labor-intensive but critical for model quality.

**Document Collection Strategy**

The quality and diversity of your training dataset significantly influences model performance:

1. Gather a representative sample of at least 200-500 documents that reflect the full range of variations you expect to encounter in production:
   * Documents from different recording offices
   * Various document types (deeds, mortgages, liens, easements)
   * Documents with different formats, layouts, and structures
   * Documents of varying quality, including older records with degraded image quality
2. Include edge cases that represent challenging scenarios your system will encounter:
   * Documents with handwritten components
   * Historical documents with archaic language
   * Documents with stamps or notary seals overlapping text
   * Documents with multiple amendments or corrections
3. Balance your dataset to prevent model bias toward particular document formats or jurisdictions.

**Document Annotation Best Practices**

The annotation phase is the most critical element in successful model training:

1. Develop clear annotation guidelines that precisely define:
   * Field boundaries (what text constitutes a "legal description")
   * Handling of ambiguous cases
   * Treatment of partial or unclear information
2. Utilize Google's Document AI Workbench for annotation, which allows for:
   * Creating custom processors
   * Defining custom entity types aligned with your extraction needs
   * Establishing relationships between entities
3. Implement a multi-stage quality assurance process:
   * Initial annotation by subject matter experts familiar with land records
   * Review by a second annotator
   * Adjudication of disagreements
   * Final validation of annotation consistency
4. Maintain consistent annotation practices throughout the process, particularly when multiple annotators are involved.

**Training Configuration**

When configuring the training process:

1. Split your dataset appropriately:
   * 70-80% for training
   * 10-15% for validation
   * 10-15% for testing
2. Configure training parameters based on Google's recommendations for document processors:
   * Start with default hyperparameters
   * Adjust batch size based on document complexity
   * Set appropriate convergence criteria
3. Select the most appropriate processor type:
   * Custom document extractor for structured information
   * Form parser for standardized forms
   * Specialized processors if dealing with invoices or receipts

**Evaluation and Refinement**

Post-training evaluation provides insights for model improvement:

1. Assess model performance using appropriate metrics:
   * Field-level precision and recall
   * Entity extraction accuracy
   * Overall document processing accuracy
2. Analyze error patterns to identify systematic issues:
   * Consistently missed fields
   * Incorrect boundary identification
   * Entity misclassification
3. Implement iterative improvement:
   * Add more examples of problematic document types
   * Refine annotation guidelines
   * Retrain the model with enhanced datasets

**Implementation Considerations**

For effective deployment of your trained model:

1. Integrate pre-processing steps to optimize document quality:
   * Image enhancement for poor quality scans
   * OCR correction mechanisms
   * Page orientation normalization
2. Implement post-processing validation rules:
   * Format validation for extracted fields
   * Cross-field consistency checks
   * Confidence score thresholds for manual review
3. Establish a feedback loop for continuous improvement:
   * Collect instances of extraction errors
   * Periodically retrain with newly annotated examples
   * Monitor drift in document formats over time

**Common Challenges and Solutions**

When training Document AI for land records, anticipate these specific challenges:

1. Handling multi-page documents with dispersed information:
   * Configure the model to process full documents rather than single pages
   * Define entity relationships that span pages
2. Managing variable formatting of key information:
   * Include diverse examples of how the same information might be presented
   * Train the model to recognize contextual cues rather than rely solely on formatting
3. Extracting complex nested information:
   * Use hierarchical entity definitions for structured data like legal descriptions
   * Train separate specialized models for highly complex components

Following these methodical steps will significantly increase the likelihood of developing an effective Document AI model for extracting data from recorded documents. The investment in thorough preparation and annotation will be reflected in the accuracy and reliability of your extraction results.

**Data Types in Land Records Documents**

Land records constitute a comprehensive system of documentation that tracks ownership, rights, and interests in real property. These records, typically maintained by county recorder's offices or similar government entities, contain extensive data. Below is a thorough examination of the data elements commonly found in land records.

**Property Identification Data**

* Legal description of the property (metes and bounds, lot and block, government survey)
* Property address and physical location
* Tax parcel identification numbers
* Township, range, and section coordinates
* Subdivision information
* Lot size and dimensions
* Property boundaries

**Ownership and Transfer Information**

* Names of current and previous property owners (grantors and grantees)
* Marital status of parties
* Types of ownership (fee simple, life estate, joint tenancy, tenancy in common)
* Purchase price or consideration paid
* Transfer date
* Recording date and time
* Document reference numbers
* Chain of title information
* Signature requirements and notarization details

**Financial and Monetary Data**

* Mortgage amounts and terms
* Loan numbers and account information
* Interest rates and payment schedules
* Lender and borrower identification
* Tax assessment values
* Property tax information
* Revenue stamps indicating transfer tax payment
* Documentary stamp tax amounts

**Liens and Encumbrances**

* Tax liens (federal, state, local)
* Mechanic's liens
* Judgment liens
* Child support liens
* Construction liens
* Homeowners' association liens
* Utility liens and assessments
* Lis pendens notices (pending litigation)

**Easements and Rights of Way**

* Utility easements (electric, water, gas, telecommunications)
* Access easements
* Conservation easements
* Drainage easements
* Solar and wind rights
* Subsurface mineral rights
* Water rights and riparian claims

**Restrictive Covenants and Land Use**

* Deed restrictions
* Homeowners' association covenants
* Building setback requirements
* Use limitations
* Architectural control provisions
* Environmental restrictions
* Historic preservation designations

**Plats and Survey Information**

* Boundary survey data
* Property dimensions and measurements
* Reference monuments and markers
* Elevation data
* Flood zone information
* Topographical features
* Easement locations
* Rights of way
* Road dedications
* Common areas

**Legal Instruments and Special Circumstances**

* Power of attorney designations
* Trustee information for property held in trust
* Estate information for inherited property
* Bankruptcy notations
* Foreclosure documentation
* Quiet title actions
* Adverse possession claims
* Condemnation proceedings
* Partition actions

**Administrative and Filing Data**

* Recording fees paid
* Book and page numbers for physical records
* Document ID numbers
* Electronic recording information
* Official seals and stamps
* Certification statements
* Corrections to previously recorded documents
* Cross-reference information to related documents

**Environmental and Land Use Data**

* Environmental covenant information
* Hazardous waste designations
* Brownfield status
* Wetland designations
* Protected habitat information
* Agricultural use designations
* Conservation district information
* Zoning classifications

**Special Designations**

* Historic property status
* Affordable housing covenants
* Age-restricted community designations
* Special tax district information
* Community development district assessments
* Military installation proximity notifications
* Airport noise zone designations
* Coastal construction control lines

This comprehensive inventory represents the breadth of data potentially included in land records. The specific information contained in any given document will depend on the jurisdiction, type of transaction, and historical period in which it was recorded. Modern land records systems continue to evolve with increased digitization and standardization efforts.

Below is an **extended model** that integrates **oil/gas/mineral (OGM) interests** (and similarly **timber interests**) into the same **document-type + categories** framework you’ve already established. The key is to recognize that **“OGM” and “timber” are subject matters**, not necessarily different legal functions (like “conveyance” or “security”). You can handle them with an **additional dimension**—for example, a “SubjectMatter” list or set—while still using **DocumentType** and **DocumentCategory** for the legal form and function.

**1. Core Idea: Separate “Subject Matter” from “Document Type” and “Categories”**

1. **DocumentType**: Tells you if it’s a **Deed**, **Lease**, **Assignment**, etc.
2. **DocumentCategory**: Tells you the **legal function(s)**—Conveyance, Security, Servitudes, Change, Involuntary Liens, or Other.
3. **Subject Matter** (new dimension): Indicates **what** is being conveyed or leased—**surface estate**, **mineral estate**, **royalty interest**, **timber rights**, etc.

**Why separate them?** Because you can have, for example, a **Deed** that conveys **mineral rights**, or a **Lease** that grants **timber-cutting rights**. In both cases, it’s still a “Lease” or “Deed” from a legal standpoint (i.e., DocumentType = DEED or LEASE), and it might also be a “Conveyance” or “Servitude” from a functional standpoint—but the underlying subject matter is different.

**2. Adding a “Subject Matter” Dimension**

Let’s define a small enum or list for **SubjectMatter**:

python

Copy

from enum import Enum

class SubjectMatter(str, Enum):

SURFACE = "SURFACE" # Default if it's just ordinary real estate

MINERAL = "MINERAL" # Oil/gas/mineral rights

ROYALTY = "ROYALTY" # Royalty interest

TIMBER = "TIMBER" # Timber rights

# You can add more as needed, e.g. WATER, SOLAR, etc.

**Note**: You could make these more granular if you commonly distinguish, for instance, “Oil & Gas” from “Other Minerals.” But many offices just lump them all under “Mineral,” or call it “OGM.”

**A. One-to-Many or Many-to-Many?**

* A document might **only** convey **timber** rights or it might convey **both** surface **and** mineral rights.
* Hence, you might want **multiple** SubjectMatters in one instrument. Example: a “Mineral and Surface Deed” conveys everything.

You could store a **set** of SubjectMatter in your code (much like categories), or you could do a **many-to-many** link table in a database.

**3. Integrating Subject Matter into Your Existing Model**

Here’s a condensed Python snippet demonstrating how you can incorporate subject\_matters alongside doc\_type and categories.

**A. Expanded Data Class**

python

Copy

from dataclasses import dataclass, field

from datetime import date

from typing import Set, Optional

# (Assume DocumentType, DocumentCategory, SubjectMatter Enums exist)

@dataclass

class RecordedDocument:

document\_id: str

doc\_type: DocumentType # e.g. DEED, LEASE, ASSIGNMENT

categories: Set[DocumentCategory] = field(default\_factory=set)

subject\_matters: Set[SubjectMatter] = field(default\_factory=set)

recording\_date: date = field(default\_factory=date.today)

instrument\_number: Optional[str] = None

short\_description: Optional[str] = None

notes: Optional[str] = None

def \_\_post\_init\_\_(self):

# If categories aren’t provided, default from doc\_type as before

if not self.categories:

default\_cats = doc\_type\_to\_categories.get(self.doc\_type, set())

self.categories = set(default\_cats)

# If no subject matter is provided, it might default to SURFACE,

# or remain empty. It's up to your business logic.

# if not self.subject\_matters:

# self.subject\_matters.add(SubjectMatter.SURFACE)

**B. Examples**

1. **Mineral Deed**
   * Legally, it’s a “Deed” (DocumentType.DEED).
   * Functionally, it’s a “Conveyance” (categories = {CONVEYANCE}).
   * Subject matter is **MINERAL**.

python

Copy

mineral\_deed = RecordedDocument(

document\_id="DOC-100",

doc\_type=DocumentType.DEED, # It's a Deed

subject\_matters={SubjectMatter.MINERAL}, # Specifically conveys mineral rights

short\_description="Mineral Deed from Alice to Bob"

)

print(mineral\_deed.doc\_type) # DEED

print(mineral\_deed.categories) # {CONVEYANCE}

print(mineral\_deed.subject\_matters) # {MINERAL}

1. **Timber Lease**
   * Legally, it’s a “Lease” (DocumentType.LEASE).
   * Functionally, it’s also a “Conveyance” (of limited interest).
   * Subject matter is **TIMBER**.

python

Copy

timber\_lease = RecordedDocument(

document\_id="DOC-200",

doc\_type=DocumentType.LEASE,

subject\_matters={SubjectMatter.TIMBER},

short\_description="Timber Lease for 5 years"

)

print(timber\_lease.doc\_type) # LEASE

print(timber\_lease.categories) # {CONVEYANCE}

print(timber\_lease.subject\_matters) # {TIMBER}

1. **Oil & Gas Lease** that also includes a mortgage-like clause
   * doc\_type = LEASE, but we add categories = {CONVEYANCE, SECURITY} because it might also secure performance.
   * subject\_matters = {MINERAL} or more specifically “OIL\_GAS” if you separate that out.
2. **Deed Reserving Timber and Mineral Rights**
   * doc\_type = DEED
   * categories = {CONVEYANCE}
   * subject\_matters = {SURFACE, TIMBER, MINERAL} (maybe the deed conveys some but reserves other interests—your notes can capture the nuance).
3. **Pooling Agreement** (OGM context)
   * doc\_type might be “AGREEMENT” (under your “OTHER” category by default).
   * subject\_matters = {MINERAL}
   * If it also sets up or modifies easements, you might add category = {SERVITUDES}.

**4. Handling Reservations, Assignments, and Similar OGM/TIMBER Nuances**

1. **Reservations**: Typically found inside a deed or lease. In your model, that’s still doc\_type=DEED (or LEASE). The specific reservation is an attribute in the **notes** or **short\_description**, and the **subject\_matters** reflect the type of interest reserved (MINERAL, TIMBER, etc.).
2. **Assignments**: If it’s an **Assignment** doc (e.g., assigning a mineral lease or timber contract), then doc\_type=ASSIGNMENT with default category = {CHANGE}. The subject matter is MINERAL or TIMBER.
3. **Agreements**: Pooling agreements, unitization agreements, seismic agreements, etc., typically fall under doc\_type=AGREEMENT with category = {OTHER} by default—and subject\_matters can be {MINERAL} or {TIMBER}, depending on the resource.

**5. Putting It All Together**

With this approach:

1. You maintain a **clean, minimal** set of **DocumentType** values: e.g., DEED, LEASE, MORTGAGE, ASSIGNMENT, etc.
2. You have your **DocumentCategory** for **legal functions** (Conveyance, Security, Servitude, Involuntary\_Liens, Change, Other).
3. You add a **SubjectMatter** dimension for **OGM, timber, or surface interests**. This can be a **set**—meaning one instrument might handle **MINERAL** and **TIMBER** in the same transaction.
4. The **notes/description** fields capture any unique complexities (reservations, partial interests, etc.).

**Advantages**

* **Avoids Category Bloat**: You don’t create separate doc\_types for “Mineral Deed,” “Royalty Deed,” “Timber Deed,” etc. Instead, you say doc\_type=DEED + subject\_matters={MINERAL, TIMBER, ...}
* **Improves Search & Reporting**: You can filter by doc\_type=DEED (to see all deeds), or filter by subject\_matters containing MINERAL (to see all OGM-related docs), or filter by categories=SECURITY (to see all docs acting as security instruments).
* **Extensible**: If tomorrow you start dealing with “Water Rights” or “Solar Easements,” you can simply add a new subject matter (SubjectMatter.WATER or SubjectMatter.SOLAR) without breaking your existing schema.

**Summary**

To **properly account for oil, gas, mineral, and timber interests** (fee, lease, assignment, reservation, pooling, etc.):

1. **Retain** your existing concept of a single **DocumentType** (DEED, LEASE, etc.) + multi-**Category** classification (Conveyance, Security, Servitude, etc.).
2. **Add** a **SubjectMatter** attribute (or set of attributes) to indicate what resource(s) the instrument covers—**MINERAL**, **TIMBER**, **ROYALTY**, etc.
3. **Use** your existing fields (e.g., “notes” or “short\_description”) to capture specific references to reservations, pooling clauses, or partial interests.

This layered approach keeps the system both **organized** and **flexible**, allowing multi-dimensional searches without creating an explosion of specialized “Mineral Deed,” “Timber Deed,” “Mineral Lease,” “Timber Lease,” etc. document types.

o1

**You said:**

Great, now type this code and relavant notes in a python file that I can save. Only type code, no leading or closing message

**ChatGPT said:**

Reasoned for 9 seconds

python

Copy

from enum import Enum

from dataclasses import dataclass, field

from datetime import date

from typing import Set, Optional

# -----------------------------

# Enumerations

# -----------------------------

class DocumentCategory(str, Enum):

"""

Represents broad legal functions a recorded document can serve.

"""

CONVEYANCE = "CONVEYANCE"

SECURITY = "SECURITY"

SERVITUDES = "SERVITUDES"

INVOLUNTARY\_LIENS = "INVOLUNTARY\_LIENS"

CHANGE = "CHANGE"

OTHER = "OTHER"

class DocumentType(str, Enum):

"""

Represents the specific type of recorded document, grouped by default categories.

"""

# 1) CONVEYANCE

DEED = "DEED" # e.g. fee, mineral, tax

PATENT = "PATENT" # tax patent, USA patent

LEASE = "LEASE"

# 2) SECURITY

DEED\_OF\_TRUST = "DEED\_OF\_TRUST"

MORTGAGE = "MORTGAGE"

ASSIGNMENT\_OF\_LEASES\_AND\_RENTS = "ASSIGNMENT\_OF\_LEASES\_AND\_RENTS"

# 3) SERVITUDES

CCRS = "CCRS" # Covenants, Conditions & Restrictions

PROTECTIVE\_COVENANTS = "PROTECTIVE\_COVENANTS"

EASEMENT = "EASEMENT"

RIGHT\_OF\_WAY = "RIGHT\_OF\_WAY"

# 4) INVOLUNTARY LIENS

CONSTRUCTION\_LIEN = "CONSTRUCTION\_LIEN" # Mechanic's Lien

FEDERAL\_TAX\_LIEN = "FEDERAL\_TAX\_LIEN"

JUDGMENT = "JUDGMENT"

LIS\_PENDENS = "LIS\_PENDENS"

UCC = "UCC"

HOA\_ASSESSMENT\_LIEN = "HOA\_ASSESSMENT\_LIEN"

# 5) CHANGE

RELEASE = "RELEASE"

PARTIAL\_RELEASE = "PARTIAL\_RELEASE"

MODIFICATION\_AGREEMENT = "MODIFICATION\_AGREEMENT"

ASSIGNMENT = "ASSIGNMENT"

SUBORDINATION\_AGREEMENT = "SUBORDINATION\_AGREEMENT"

SUBSTITUTION\_OF\_TRUSTEE = "SUBSTITUTION\_OF\_TRUSTEE"

# 6) OTHER

COURT\_DECREE = "COURT\_DECREE"

AFFIDAVIT = "AFFIDAVIT"

AGREEMENT = "AGREEMENT"

POWER\_OF\_ATTORNEY = "POWER\_OF\_ATTORNEY"

TRUST\_AGREEMENT = "TRUST\_AGREEMENT"

PLATS = "PLATS"

PLANS = "PLANS"

class SubjectMatter(str, Enum):

"""

Represents the resource or interest that is the subject of the document.

Useful for oil, gas, mineral, timber, etc.

"""

SURFACE = "SURFACE"

MINERAL = "MINERAL"

ROYALTY = "ROYALTY"

TIMBER = "TIMBER"

# -----------------------------

# Default Category Mapping

# -----------------------------

doc\_type\_to\_categories = {

# CONVEYANCE

DocumentType.DEED: {DocumentCategory.CONVEYANCE},

DocumentType.PATENT: {DocumentCategory.CONVEYANCE},

DocumentType.LEASE: {DocumentCategory.CONVEYANCE},

# SECURITY

DocumentType.DEED\_OF\_TRUST: {DocumentCategory.SECURITY},

DocumentType.MORTGAGE: {DocumentCategory.SECURITY},

DocumentType.ASSIGNMENT\_OF\_LEASES\_AND\_RENTS: {DocumentCategory.SECURITY},

# SERVITUDES

DocumentType.CCRS: {DocumentCategory.SERVITUDES},

DocumentType.PROTECTIVE\_COVENANTS: {DocumentCategory.SERVITUDES},

DocumentType.EASEMENT: {DocumentCategory.SERVITUDES},

DocumentType.RIGHT\_OF\_WAY: {DocumentCategory.SERVITUDES},

# INVOLUNTARY LIENS

DocumentType.CONSTRUCTION\_LIEN: {DocumentCategory.INVOLUNTARY\_LIENS},

DocumentType.FEDERAL\_TAX\_LIEN: {DocumentCategory.INVOLUNTARY\_LIENS},

DocumentType.JUDGMENT: {DocumentCategory.INVOLUNTARY\_LIENS},

DocumentType.LIS\_PENDENS: {DocumentCategory.INVOLUNTARY\_LIENS},

DocumentType.UCC: {DocumentCategory.INVOLUNTARY\_LIENS},

DocumentType.HOA\_ASSESSMENT\_LIEN: {DocumentCategory.INVOLUNTARY\_LIENS},

# CHANGE

DocumentType.RELEASE: {DocumentCategory.CHANGE},

DocumentType.PARTIAL\_RELEASE: {DocumentCategory.CHANGE},

DocumentType.MODIFICATION\_AGREEMENT: {DocumentCategory.CHANGE},

DocumentType.ASSIGNMENT: {DocumentCategory.CHANGE},

DocumentType.SUBORDINATION\_AGREEMENT: {DocumentCategory.CHANGE},

DocumentType.SUBSTITUTION\_OF\_TRUSTEE: {DocumentCategory.CHANGE},

# OTHER

DocumentType.COURT\_DECREE: {DocumentCategory.OTHER},

DocumentType.AFFIDAVIT: {DocumentCategory.OTHER},

DocumentType.AGREEMENT: {DocumentCategory.OTHER},

DocumentType.POWER\_OF\_ATTORNEY: {DocumentCategory.OTHER},

DocumentType.TRUST\_AGREEMENT: {DocumentCategory.OTHER},

DocumentType.PLATS: {DocumentCategory.OTHER},

DocumentType.PLANS: {DocumentCategory.OTHER},

}

# -----------------------------

# Main Data Model

# -----------------------------

@dataclass

class RecordedDocument:

"""

Represents a recorded document with a single DocumentType,

one or more categories, and optional subject matters like minerals or timber.

"""

document\_id: str

doc\_type: DocumentType

categories: Set[DocumentCategory] = field(default\_factory=set)

subject\_matters: Set[SubjectMatter] = field(default\_factory=set)

instrument\_number: Optional[str] = None

recording\_date: date = field(default\_factory=date.today)

short\_description: Optional[str] = None

notes: Optional[str] = None

def \_\_post\_init\_\_(self):

# If categories are not explicitly set, derive from doc\_type

if not self.categories:

default\_cats = doc\_type\_to\_categories.get(self.doc\_type, set())

self.categories = set(default\_cats)