Foundations

Multimodal Fundamentals

PDFs, Layout, Tables/Charts, Images, and Audio

Michael J Bommarito II · Jillian Bommarito · Daniel Martin Katz

December 21, 2025

Working Draft Chapter

Version 0.1

Practical multimodal workflows for legal and finance: preserve structure, extract tables, summarize audio, and protect privacy.

Contents

How to Read This Chapter

Focus on the sections aligned to your data: PDFs and layout for filings/contracts, tables/charts for financials, audio/transcripts for calls.

Key Objectives

- Preserve document structure and identifiers (pages, figures, tables).
- Extract tables/charts accurately and capture units/footnotes.
- Summarize audio with timestamps and speaker labels.
- Apply privacy/redaction safeguards before external processing.

0.1 Introduction and Scope

Retrieval-augmented generation has evolved from simple text chunking to sophisticated **multimodal RAG** pipelines capable of processing complex documents, images, and rich media. The ‘‘text-only’’ RAG pipeline is increasingly seen as legacy architecture, insufficient for enterprise data locked in PDFs, charts, slide decks, and recorded proceedings.

This chapter addresses practical multimodal inputs common to legal and finance: preserving structure in PDFs, extracting tables and charts, interpreting images and screenshots, and summarizing audio and video. We also highlight privacy and redaction patterns that must be applied before sensitive content enters AI systems.

0.1.1 The Multimodal Imperative

Legal and financial workflows are inherently multimodal:

- **SEC filings**: Combine narrative text with financial tables, charts, and embedded images.
- **Contracts**: May include scanned signatures, attached exhibits, and referenced schedules.
- **Litigation materials**: Span depositions (audio/video), exhibits (images, documents), and transcripts.
- **Regulatory correspondence**: Often arrives as scanned PDFs requiring OCR.
- **Research reports**: Integrate text analysis with data visualizations.

A system that can only process plain text misses critical information encoded in these other modalities---and may produce incomplete or misleading analysis.

0.1.2 Building on Prior Foundations

This chapter builds directly on concepts from earlier chapters:

- **Embeddings** (Chapter 1): Vector representations extend beyond text to images, tables, and audio transcripts. Models like CLIP project multiple modalities into shared embedding spaces.
- **Structured outputs** (Chapter 3): Extracted tables and metadata should conform to defined schemas for downstream integration.
- **Tool use** (Chapter 3): Document parsers, OCR engines, and ASR models act as tools that preprocessing agents invoke.
- **Evidence records** (Chapter 3): Multimodal processing steps---redaction, extraction, transcription---must be logged with the same rigor as LLM inference.

Chapter Scope

We focus on *ingestion and preprocessing*---getting multimodal content into a form suitable for LLM processing and retrieval. Generation of multimodal outputs (images, audio synthesis) is beyond our current scope, though content authenticity standards apply to both directions.

0.2 Document Structure and Layout

The primary bottleneck in enterprise document workflows is the Portable Document Format (PDF). Standard text extraction tools treat a PDF as a stream of text characters, often destroying the semantic structure of tables, multi-column layouts, and headers. This ‘‘PDF problem’’ results in systems that

can retrieve text but cannot understand the relationship between a data point in a table cell and its row/column headers---a critical failure mode for legal and financial analysis.

0.2.1 The Spectrum of Document Parsing Strategies

Modern document parsing spans a spectrum from simple text extraction to sophisticated visual understanding:

Document Parsing Strategies

Text Stream (Legacy) Extract characters via libraries like `pypdf`. Fast and cheap, but destroys tables, columns, and reading order.

Heuristic Parsing Rule-based approaches using whitespace and line detection (e.g., `pdfplumber`). Better table support but brittle on borderless or complex tables.

Layout Models (AI) Neural models like **LayoutLM** or **DocLayout-YOLO** that identify structural elements through bounding boxes. Identifies headers versus body text but requires GPU resources.

Vision-First (VLM) Screenshot pages and process through vision-language models like GPT-4V. ''Human-like'' understanding of layout but slow, expensive, and limited by context windows.

0.2.2 Layout Analysis Models

To solve the PDF problem, modern pipelines employ **layout analysis models** that treat the document page as an image---or a hybrid of image and text box coordinates---to identify visual blocks: headers, paragraphs, tables, and figures.

LayoutLM and Successors.. Microsoft's LayoutLM family combines text understanding with spatial awareness. The model receives both the OCR text and the bounding box coordinates for each text segment, allowing it to learn the relationship between content and position. LayoutLMv3 and subsequent models add visual features from the document image itself, enabling recognition of logos, signatures, and other non-textual elements.

DocLayout-YOLO.. For high-throughput pipelines, object detection architectures like YOLO (''You Only Look Once'') have been adapted for document layout analysis. DocLayout-YOLO processes document images in a single forward pass, detecting and classifying regions as headers, paragraphs, tables, figures, or footnotes. This approach trades some accuracy for significantly faster processing.

Azure Document Intelligence.. Microsoft's cloud service provides pre-trained models for common document types (invoices, receipts, contracts) and allows custom model training for specialized

formats. The service returns structured JSON with identified fields, tables, and key-value pairs, along with confidence scores and bounding polygons.

0.2.3 Preserving Structure for Downstream Use

Regardless of the parsing approach, the goal is to preserve sufficient structure for downstream tasks:

Structure Preservation Checklist

- **Reading order**: Reconstruct the logical sequence for multi-column layouts.
- **Hierarchy**: Preserve heading levels, section numbering, and nested lists.
- **Object identifiers**: Retain figure/table numbers and their captions.
- **Page references**: Record page numbers for precise citations.
- **Footnotes/endnotes**: Link footnote markers to their content.
- **Cross-references**: Preserve internal document links where possible.

For legal filings and financial disclosures, page numbers and exhibit references are essential for citation. A contract review system that cannot identify ‘‘Section 4.2(a)’’ or ‘‘Exhibit B’’ loses the precision that practitioners require.

0.2.4 Chunking Strategies for Retrieval

Once structure is identified, documents must be divided into chunks for embedding and retrieval. Naive approaches split by token count, but structure-aware chunking yields better results:

Structure-Aware Chunking

Semantic boundaries Split at section headings rather than arbitrary token counts.

Table isolation Keep tables as atomic units with their captions.

Contextual overlap Include section headers in each chunk for context.

Metadata preservation Attach page numbers, section paths, and document identifiers to each chunk.

For financial documents, a common pattern is to extract tables into a separate structured index while chunking the narrative text. This allows the retrieval system to search both modalities and combine results during synthesis.

0.3 Tables and Charts

Tables and charts represent high-density information that requires specialized handling. A financial statement table, regulatory schedule, or litigation exhibit contains relationships between cells, headers, and footnotes that simple text extraction destroys. Similarly, charts encode trends and comparisons visually that cannot be reconstructed from OCR alone.

0.3.1 Table Extraction Strategies

Table extraction has evolved from rule-based approaches to sophisticated vision-language methods:

Heuristic Parsers.. Libraries like `pdfplumber` and `Camelot` detect tables through line detection and whitespace analysis. These work well for clean, bordered tables but fail on borderless tables, merged cells, or complex headers common in legal and financial documents.

Vision-Based Extraction.. State-of-the-art approaches utilize vision-language models (VLMs) to parse tables. Instead of reconstructing the table from text coordinates, the system sends an image of the table to a VLM with a prompt to ''transcribe this table to Markdown'' or ''convert to HTML.'' This preserves merged cells and complex headers far better than heuristic parsers.

Vision-Based Table Extraction Prompt

```
Extract this table to Markdown format.
Preserve:
- All merged cells (span them appropriately)
- Header hierarchy (multi-row headers)
- Footnote markers (superscript numbers)
- Currency symbols and units
Return only the Markdown table, no explanation.
```

Structured Output for Tables.. When tables must integrate with downstream systems, request structured output:

- **JSON with metadata**: Include row/column headers, units, footnotes, and source page numbers.
- **CSV with context**: Preserve column types and include a header comment with table caption.
- **HTML tables**: Maintain cell spans and styling for complex structures.

0.3.2 Chain-of-Table Reasoning

For reasoning over tables, the **Chain-of-Table** framework dynamically plans operations to navigate a table. Rather than ingesting the whole table into the context, the model iteratively generates operations to create a virtual, simplified table that answers the specific query.

Chain-of-Table Operations

Filter Select rows matching criteria: ‘‘Year > 2020’’

Select Choose specific columns: ‘‘Revenue, Net Income’’

Aggregate Compute sums, averages, or counts

Sort Order by column values

Join Combine with another table on a key column

This mimics how an analyst works with a spreadsheet---progressively narrowing the data to answer a question rather than overwhelming the model with the entire table.

0.3.3 Chart Understanding

Charts are often ignored in text-based retrieval systems, yet they encode critical information in legal and financial documents: stock price trends, market share comparisons, revenue projections.

The CHARGE Framework.. The Chart-based Question Answering Generation (CHARGE) framework extracts keypoints from charts and verifies them against the text to generate QA pairs. This ensures the model can ‘‘read’’ the data trends visually represented in the document, allowing users to ask questions like ‘‘What was the trend in Q3 according to the bar chart?’’

Chart-to-Table Conversion.. One practical approach converts charts to structured data:

1. Send the chart image to a VLM with instructions to extract the underlying data.
2. Request output as a table (CSV or JSON) with axis labels and values.
3. Store both the extracted data and a link to the original chart image.
4. During synthesis, the model can reference either the extracted data or describe the visual.

Chart Extraction Limitations

Be cautious when reconstructing data from charts:

- **Precision loss**: Values read from axis positions are approximate.
- **Missing data points**: Not all values may be visible or labeled.
- **Scale ambiguity**: Logarithmic or truncated axes affect interpretation.
- **Prefer originals**: When available, use the source data files rather than chart extraction.

0.3.4 Multimodal Embeddings for Tables and Charts

Once content is extracted, it must be indexed for retrieval. Two architectural approaches dominate:

Unified Embeddings.. Models like **CLIP** (Contrastive Language-Image Pre-training) and **SigLIP** project images and text into a shared vector space. This allows cross-modal retrieval: a user can type a text query (''Show me the graph of rising interest rates'') and retrieve the relevant image from a slide deck.

Late Fusion.. A robust architecture often employs **late fusion**. Instead of a single embedding space, the system maintains separate indices for text and images (using specialized models for each). During retrieval, candidates are fetched from both indices, and a re-ranking model fuses the scores to present the most relevant mixed-media results.

When to Use Each Approach

Unified embeddings Best for general cross-modal search where text and images are loosely related.

Late fusion Preferred when the nuance of a specific modality (e.g., OCR text inside an image) is critical, or when you need fine-grained control over retrieval weights.

For financial documents with complex tables, late fusion often outperforms unified embeddings because it can leverage specialized table understanding models alongside general-purpose text embedders.

0.4 Audio and Transcripts

Extending retrieval-augmented generation to temporal media (audio and video) introduces the dimension of time. A retrieved result is not just a ''document'' but a specific time span within a media file. For legal and financial practitioners, this means earnings calls, depositions, regulatory hearings, and training recordings become searchable and quotable with timestamp precision.

0.4.1 Audio RAG Pipelines

Audio RAG pipelines depend on the quality of automatic speech recognition (ASR) and the preservation of temporal metadata:

Transcription with Timestamps.. Models like **Whisper** (OpenAI) and **AssemblyAI** convert audio to text while preserving word-level or segment-level timestamps. When a relevant chunk is found during retrieval, the system maps the text back to the original timestamps, allowing the user to ''jump to'' the exact moment in the audio player.

Speaker Diarization.. Crucially, the transcription step must include **speaker diarization**---identifying who is speaking. ‘‘Speaker A said X’’ is semantically different from ‘‘Speaker B said X.’’ In a deposition or earnings call, attributing statements to the correct speaker is essential for accurate analysis.

Audio RAG Pipeline Components

1. **Ingestion**: Audio files processed through ASR with diarization enabled.
2. **Segmentation**: Text chunked by semantic breaks, speaker turns, or silence rather than arbitrary token counts.
3. **Embedding**: Transcript segments embedded with speaker and timestamp metadata.
4. **Retrieval**: Query matches return text plus temporal coordinates.
5. **Synthesis**: Response includes citations with timestamps and optional audio playback links.

Error Rates and Mitigation.. ASR is imperfect. Technical terminology, proper names, and accented speech increase word error rates (WER). For legal and financial applications:

- Provide custom vocabularies (company names, legal terms) to the ASR system.
- Consider human review for high-stakes transcripts (depositions, regulatory testimony).
- Retain the original audio alongside transcripts for verification.
- Display confidence scores where available to flag uncertain passages.

0.4.2 Video Understanding and Retrieval

Video RAG treats video as a sequence of visual frames synchronized with an audio track, enabling queries that span both modalities.

Dual-Stream Indexing.. A comprehensive video RAG system indexes both:

- **Transcript vectors**: What was said (from ASR with diarization).
- **Visual frame descriptions**: What was shown (from keyframe extraction and VLM captioning).

A user query searches both streams, allowing questions like ‘‘Find the scene where the speaker discusses quarterly revenue while showing the bar chart.’’

Keyframe Extraction.. Keyframes are extracted at regular intervals (e.g., 1 frame per second) or at scene changes. Each frame is processed by a VLM to generate textual descriptions (‘‘scene

graphs'') or embedded directly using CLIP. For legal and financial video---training materials, recorded presentations, regulatory hearings---meaningful frames often coincide with slide transitions.

VideoRAG Architecture.. Advanced frameworks like **VideoRAG** employ a dual-channel architecture with ''Graph-based Textual Knowledge Grounding'' to transform visual signals into structured text representations while preserving temporal dependencies. This allows complex queries that span both audio and visual content.

Multimodal Video Query

Query: ''Find where the CFO discusses the accounting change while the slide shows the impact table.''
System behavior:

1. Search transcript for ''accounting change'' + speaker ''CFO''
2. Search visual index for ''table'' or ''impact''
3. Intersect temporal windows to find overlapping segments
4. Return video clips with start/end timestamps

0.4.3 Practical Considerations

Storage and Streaming.. Video and audio files are large. Systems typically:

- Store original media in object storage (S3, Azure Blob).
- Generate and index transcripts/descriptions separately.
- Stream relevant segments via FFMPEG or cloud media services.
- Return playback links with timestamp parameters rather than downloading entire files.

Privacy and Access Control.. Audio and video often contain sensitive content---voices are biometric identifiers, and recordings may capture privileged communications. Apply the privacy controls discussed in Section 0.5 before ingestion:

- Redact or exclude segments containing privileged discussions.
- Apply speaker-level access controls where content is speaker-specific.
- Consider whether transcripts alone (without audio) suffice for the use case.

Audio/Video RAG Best Practices

- Always preserve timestamp-to-text mappings for citation.
- Enable speaker diarization for multi-party recordings.
- Provide custom vocabularies for domain-specific terminology.
- Retain original media for verification of AI-generated transcripts.
- Apply access controls at the segment level where sensitivity varies.

0.5 Privacy and Redaction

Data leakage is a primary concern in multimodal RAG systems, while misinformation is a concern in generation. Before data enters the vector database or the LLM context window, it must be scrubbed of personally identifiable information (PII), privileged content, and other sensitive material. Equally important, the provenance of AI-generated content must be trackable.

0.5.1 PII Detection and Redaction

The Challenge.. Documents processed through multimodal pipelines often contain:

- **Direct identifiers**: Names, Social Security numbers, account numbers.
- **Quasi-identifiers**: Dates, locations, and demographic details that enable re-identification.
- **Sensitive categories**: Health information, financial data, legal case details.
- **Embedded PII**: Information within images, scanned forms, or handwritten notes.

Microsoft Presidio.. Presidio is an open-source framework for detecting, redacting, masking, and anonymizing sensitive data. It combines:

- **Pattern matching**: Regular expressions for structured identifiers (SSN, phone numbers, credit cards).
- **Named entity recognition (NER)**: Machine learning models to identify names, organizations, and locations.
- **Configurable anonymizers**: Replace, mask, hash, or encrypt detected entities.
- **Extensibility**: Custom recognizers for domain-specific identifiers (case numbers, account formats).

Presidio Redaction Pipeline

A document processing pipeline might:

1. Extract text from PDF using layout analysis.
2. Pass text through Presidio's analyzer to detect PII entities.
3. Apply anonymizers: replace names with tokens, mask account numbers.
4. Embed and index the redacted text.
5. Store mapping between tokens and original values in a secure vault (if reversible redaction is needed).

Image-Based PII.. For scanned documents and images:

- Run OCR to extract text, then apply text-based PII detection.
- Use bounding box coordinates to redact regions in the original image.
- Consider visual PII (faces, signatures) that text-based methods miss.
- Presidio Image Redactor extends the framework to handle images directly.

0.5.2 Privilege and Confidentiality

Beyond PII, legal and financial workflows must protect:

- **Attorney-client privilege**: Communications protected from disclosure.
- **Work product doctrine**: Attorney mental impressions and legal strategy.
- **Trade secrets**: Proprietary business information.
- **Material non-public information (MNPI)**: Information that could affect securities prices.

Privilege in AI Systems

Including privileged content in a shared vector database or sending it to an external LLM API may waive privilege. Design systems to:

- Segregate privileged content into separate indices with strict access controls.
- Use on-premises or private cloud deployments for sensitive processing.
- Implement privilege review workflows before ingestion.
- Log all access to privileged content for audit purposes.

0.5.3 Content Authenticity and Provenance

As AI generates increasingly realistic content, tracking provenance becomes critical. The **Content Authenticity Initiative (CAI)** and **C2PA** (Coalition for Content Provenance and Authenticity) standards address this need.

Content Credentials.. C2PA enables cryptographic signing of media files. This metadata (''Content Credentials'') travels with the file, proving:

- **Origin**: Whether content was AI-generated, camera-captured, or edited.
- **Editing history**: What modifications were applied and by whom.
- **Tool chain**: Which software or AI models were involved.

This provides a ''digital nutrition label'' that allows consumers to verify the provenance of the content they are viewing.

Application to Legal and Finance.. Content Credentials are particularly relevant for:

- **Evidence authenticity**: Establishing the chain of custody for digital evidence.
- **AI-generated disclosures**: Marking synthetic content in regulatory filings.
- **Document integrity**: Proving that a contract or filing has not been tampered with.
- **Audit trails**: Demonstrating the provenance of AI-assisted analysis.

0.5.4 Redaction Governance

Effective redaction requires governance beyond the technical implementation:

Redaction Governance Checklist

- **Policy documentation**: Define what must be redacted and under what circumstances.
- **Version control**: Track changes to redaction rules over time.
- **Exception handling**: Document when and why redaction was overridden.
- **Audit logging**: Record who performed redactions, when, and what was affected.
- **Reversibility decisions**: Determine if redaction should be reversible and secure the mapping.
- **Quality assurance**: Sample and review redacted output for completeness.

Integration with Evidence Records.. Redaction events should be captured in the canonical evidence record (see Chapter 3). When a document is processed:

- Log the redaction rules applied (version, configuration).
- Record entities detected and actions taken.
- Preserve checksums of both original and redacted content.
- Link to the redaction policy governing the action.

This ensures that any downstream analysis can be traced back to the original data with full understanding of what was removed and why.

0.6 Synthesis

Multimodal RAG represents the maturation of retrieval-augmented generation from a text-processing technique to a comprehensive perception system. By addressing document structure, tables and charts, audio and video, and privacy safeguards, you can build workflows that match the multimodal reality of legal and financial practice.

0.6.1 Integration Patterns

The components discussed in this chapter work together in layered pipelines:

1. **Ingestion layer**: Documents enter the system and are classified by type (PDF, image, audio, video).
2. **Preprocessing layer**: Layout analysis, table extraction, OCR, and ASR transform raw content into structured text with metadata.
3. **Privacy layer**: PII detection and redaction sanitize content before it enters shared indices or external APIs.
4. **Embedding layer**: Content is vectorized---potentially through multiple specialized embedders (text, image, table).
5. **Indexing layer**: Vectors and metadata are stored with provenance information.
6. **Retrieval layer**: Queries search across modalities, with late fusion combining results.
7. **Synthesis layer**: Retrieved content is presented to the LLM with appropriate context for generation.

0.6.2 Key Decisions

When designing multimodal pipelines, key architectural decisions include:

- **Parsing strategy**: Heuristic, AI-based layout models, or vision-first (VLM)?
- **Embedding architecture**: Unified multimodal embeddings or late fusion?
- **Privacy approach**: Pre-ingestion redaction, access controls, or both?
- **Media handling**: Stream from source or cache processed segments?
- **Provenance depth**: Minimal logging or full W3C PROV-O lineage?

The right answers depend on your accuracy requirements, latency constraints, cost sensitivity, and regulatory obligations.

Multimodal Perception in Agentic Systems

Document processing and multimodal understanding become *perception capabilities* in agentic systems. Chapter 7's Perception question addresses how agents access and interpret diverse information sources, while the Governance question covers privacy controls and data isolation requirements. The pipelines described here form the sensory apparatus through which agents perceive their documentary environment.

0.6.3 Looking Forward

With multimodal ingestion in place, the next challenge is designing prompts and evaluation frameworks that leverage these capabilities effectively. Chapter 5 treats prompt design, strategy selection, evaluation, and optimization as an engineering discipline---applying structured thinking to the interface between human intent and model behavior.

0.7 Further Learning

0.7.1 Document Layout and Structure

- **LayoutLM**: Xu et al. introduced LayoutLM for document understanding, combining text and layout information in a pre-trained model. The LayoutLMv3 paper extends this with unified text-image pre-training.
- **DocLayout-YOLO**: Adapts object detection for document layout analysis, enabling fast identification of headers, paragraphs, tables, and figures.
- **Azure Document Intelligence**: Microsoft's documentation provides practical guidance on

layout analysis, table extraction, and custom model training for enterprise document types.

- **SCAN**: Recent work on Semantic Document Layout Analysis for visual and textual RAG pipelines.

0.7.2 Table and Chart Understanding

- **Chain-of-Table**: Wang et al. (ICLR 2024) present a framework for table reasoning through iterative operations rather than full table ingestion.
- **CHARGE**: The Chart-based Question Answering Generation framework for extracting and verifying information from data visualizations.
- **Vision-Based Table Extraction**: Elastic Search Labs and others document practical approaches for parsing PDF tables using VLMs.

0.7.3 Audio and Video RAG

- **Whisper**: OpenAI's open-source ASR model supports multiple languages with word-level timestamps.
- **VoxRAG**: Research on transcription-free RAG systems for spoken question answering.
- **VideoRAG**: Frameworks for long-context video understanding with dual-channel (audio + visual) retrieval.
- **Speaker Diarization**: AssemblyAI and pyannote.audio provide practical tools for identifying speakers in multi-party recordings.

0.7.4 Multimodal Embeddings

- **CLIP**: OpenAI's Contrastive Language-Image Pre-training model enables cross-modal search between text and images.
- **SigLIP**: Google's Sigmoid Loss for Language Image Pre-Training improves on CLIP for certain retrieval tasks.
- **Late Fusion Architectures**: Research comparing unified embeddings versus modality-specific indices with score fusion.

0.7.5 Privacy and Content Authenticity

- **Microsoft Presidio**: Open-source PII detection and anonymization framework with support for text and images.
- **C2PA**: The Coalition for Content Provenance and Authenticity specification for cryptographic content credentials.

- **Content Authenticity Initiative**: Adobe-led consortium developing tools and standards for content provenance.

Conclusion

The transition from text-only to multimodal RAG represents a qualitative shift in what AI systems can perceive and process. By applying layout analysis to documents, specialized extraction to tables and charts, ASR with diarization to audio and video, and rigorous privacy controls throughout, you can build systems that match the multimodal reality of legal and financial practice.

Key takeaways from this chapter:

- **Structure matters**: The ‘‘PDF problem’’ destroys semantic relationships. Layout analysis models preserve the structure that practitioners need for accurate citation and analysis.
- **Tables require special handling**: Whether through heuristic parsers, vision-based extraction, or Chain-of-Table reasoning, tables must be treated as first-class objects, not flattened text.
- **Time is a dimension**: Audio and video RAG adds temporal coordinates to retrieval, enabling precise citation of spoken content.
- **Privacy is non-negotiable**: PII detection, redaction, and privilege protection must occur before content enters shared systems or external APIs.
- **Provenance enables trust**: Content Credentials and evidence records establish the chain of custody for AI-processed content.

With multimodal ingestion capabilities in place, the next challenge is the human-AI interface: how do we communicate intent to these systems effectively? Chapter 5 addresses prompt design, evaluation, and optimization as engineering disciplines---applying the same rigor to the interface layer that we have applied to structured outputs, tool use, and multimodal processing.