RMS. DEEP TECH REPORT & WHITEPAPER

Advanced RAG: Architekturen zur präzisen Verarbeitung von Tabellen, Grafiken und Rechnungen in Enterprise-Systemen

Datum: 15.06.2026
Autoren: Stefanie Fink, Michael Kettel, Marco Schmidt
Herausgeber: rms. relationship marketing solutions GmbH, Stuttgart

In modernen Unternehmensumgebungen ist die automatisierte Beantwortung von Fragen aus internen Dokumenten mittels Retrieval-Augmented Generation (RAG) zu einem zentralen Effizienzfaktor geworden. Während textlastige Dokumente durch traditionelle Vektor-Pipelines zuverlässig verarbeitet werden können, scheitern Standard-Systeme systematisch an geschäftskritischen Dokumentstrukturen wie verschachtelten Tabellen, komplexen Grafiken und strukturierten Rechnungen. Dieses Whitepaper beschreibt die technischen Herausforderungen im Detail und stellt produktionsbereite Architektur-Frameworks zur Implementierung von **Advanced Multimodal RAG** vor.

1. Das strukturelle Defizit naiver RAG-Systeme

Die mathematische Abbildung von Textabschnitten in dichten Vektoren (***Dense Embeddings***) basiert auf sequenzieller Semantik. Typische textbasierte Chunker segmentieren Dokumente nach rein quantitativen Kriterien wie der Token-Anzahl oder festen Zeichenlängen mit Überlappung. Trifft ein solcher Algorithmus auf eine Tabelle oder eine visuell strukturierte Rechnung, bricht die logische Kontiguität zusammen. Der Grund liegt in der Reduzierung einer zweidimensionalen Beziehung (Zeile × Spalte) auf einen eindimensionalen Textstrom.

Erhält ein Large Language Model (LLM) während des Generierungsprozesses lediglich isolierte Tabellenzeilen, die als unstrukturierter Fließtext ohne Spaltenüberschriften extrahiert wurden, verliert der interne Aufmerksamkeitsmechanismus (***Attention Mechanism***) die mathematische Zuordnung. Ein Fragment wie `"4.200 | 12.850 | 1.100"` besitzt im dichten Vektorraum zwar eine hohe Kosinus-Ähnlichkeit zu vielen numerischen Kontexten, verliert jedoch jegliche semantische Trennschärfe darüber, ob es sich um Umsatzzahlen, Artikelnummern oder physikalische Einheiten handelt. Das Resultat in produktiven Systemen sind fehlerhafte statistische Aggregatdaten und unvorhersehbare Halluzinationen.

2. Strategische Lösungsarchitekturen für komplexe Dokumente

A. Visuelles Retrieval und Late Interaction (Das ColPali-Paradigma)

Der fortschrittlichste Deep-Tech-Ansatz verabschiedet sich vollständig von der fehleranfälligen optischen Zeichenerkennung (OCR) auf Dokumentenebene und verlagert das Information Retrieval direkt auf die visuelle Achse. Kern dieser Architektur ist **ColPali**, eine Erweiterung von Vision-Language-Modellen (VLMs) wie PaliGemma.

- **Visuelle Indexierung:** Jede PDF-Seite wird als Bildmatrix durch einen Vision-Encoder (z. B. SigLIP) geschleust. Statt eines einzigen globalen Vektors erzeugt das Modell eine Vielzahl von Token-Vektoren für einzelne Bildsegmente (***Patches***).
- **Late Interaction Mechanismus:** Bei einer Suchanfrage wird die Text-Query des Nutzers durch das Sprachmodell projiziert. Die Relevanzberechnung erfolgt über eine ***MaxSim***-Operation, die jedes Query-Token mit dem visuell stärksten Bild-Patch abgleicht. Dadurch werden Tabellenköpfe, fettgedruckte Summenlinien oder Achsenbeschriftungen in Grafiken direkt über ihre geometrische Anordnung im Raum indiziert.
- **Multimodale Injektion:** Das System extrahiert bei einem Treffer nicht den Text, sondern übergibt die hochauflösende Bildmatrix der Seite direkt an ein generatives VLM, welches die visuelle Struktur nativ analysiert.

B. Multi-Vector-Retrieval und syntaktische Repräsentationen

Für Infrastrukturen, die primär auf reine Text-LLMs angewiesen sind, erfordert die Pipeline eine strikte Entkopplung von Suchkontext und Generierungskontext.

Mithilfe neuronaler Layout- und Parsing-Modelle (z. B. der rms.-Pipeline aus Kreuzberg-PDF, Tesseract und PaddleOCR) wird das Dokument in logische Funktionseinheiten partitioniert. Tabellen und Diagramme werden deterministisch isoliert. Jede isolierte Entität wird anschließend an ein kompaktes Vision-Modell übergeben, welches eine präzise textuelle Zusammenfassung aller numerischen Werte und Spaltenbeschriftungen generiert.

In der Vektordatenbank wird der dichte Vektor dieser Zusammenfassung für die semantische Suche hinterlegt. Als Nutzlast (***Payload***) wird jedoch die exakt konvertierte Tabelle im HTML-Format oder als strukturiertes Markdown gespeichert. Wird die Beschreibung durch die Suchanfrage getroffen, injiziert das System die strukturelle Tabellenrepräsentation in den Prompt des LLMs.

> **rms. Enterprise-Standard: Tabellen-Injektion via HTML**
>
> Untersuchungen im rms. LLM-Lab zeigen, dass generative Sprachmodelle das native HTML-Tabellenformat (`<table>`, `<tr>`, `<td>`) mit einer um bis zu 32 % höheren strukturellen Präzision verarbeiten als CSV-Formate. Da die vortrainierten Basismodelle riesige Mengen an Web-Dokumenten analysiert haben, bleibt die semantische Verankerung zwischen Spalten-Header und Zellenwert über HTML-Tags mathematisch robuster erhalten.

C. Hybrid-Search und deterministisches Metadaten-Filtering für Invoices

Bei Finanzdokumenten und Rechnungen kollidiert die Unschärfe semantischer Vektoren mit der erforderlichen deterministischen Exaktheit von Finanzprozessen. Eine Suche nach Rechnungsbeträgen darf keine statistische Näherung sein. Die Systemarchitektur muss daher zwingend hybrid aufgebaut sein. Jede eingehende Rechnung durchläuft eine Extraktion, die Schlüssel-Wert-Paare in ein striktes JSON-Schema überführt. Die Vektordatenbank speichert das Dokument und verknüpft den semantischen Vektor des Freitextes mit strukturierten Skalar-Indizes:

```
{
  "document_id": "INV-2026-RMS-084",
  "payload": {
    "vendor": "Solarize GmbH",
    "total_amount": 1428.50,
    "tax_rate": 19.00,
    "year": 2026,
    "line_items": ["API-Schnittstelle", "Supportbot-Integration"]
  },
  "vector_dense": [0.0412, -0.2189, 0.8431, ...]
}
```

Formuliert ein Anwender eine konkrete Anfrage, führt die Pipeline vor der eigentlichen Vektorsuche ein explizites Datenbank-Filtering aus:

$$\textit{Filter: vendor} == \textit{"Solarize GmbH"} \ \wedge \ \textit{year} == 2026$$

Die anschließende mathematische Ähnlichkeitssuche operiert ausschließlich auf der extrem eingeschränkten, exakt vorselektierten Teilmenge. False Positives werden mathematisch ausgeschlossen.

3. Post-Retrieval: Reciprocal Rank Fusion (RRF) und Cross-Encoder Reranking

Um die Ergebnisse aus der Kombination von Vektorsuche (Dense) und Keyword-Suche (Sparse/BM25) optimal zusammenzuführen, wird ein zweistufiges Post-Retrieval-Verfahren implementiert:

1. **Reciprocal Rank Fusion (RRF):** Da die Scores von BM25 und Dense Embeddings nicht direkt vergleichbar sind, bewertet der RRF-Algorithmus rein die Platzierung eines Dokuments in den jeweiligen Listen:

$$RRF\_Score(d \in D) = \Sigma_{m \in M} (1 / (k + r_m(d)))$$

Hierbei ist $r_m(d)$ der Rang des Dokuments in dem jeweiligen Suchsystem und k eine Konstante (Standard: 60).

1. **Cross-Encoder Reranking:** Die Top-Dokumente aus der RRF-Fusion werden an ein spezialisiertes Reranker-Modell (z. B. via Jina Rerank oder IONOS Qwen3-VL Reranker-8B) übergeben. Im Gegensatz zu Bi-Encodern analysiert der Cross-Encoder die Query und den Dokumentinhalt simultan über tiefe Attention-Schichten. Nur die höchstbewerteten Top-5 Dokumente werden schließlich in das Kontextfenster des Generierungs-LLMs eingespeist.

4. Validierter Enterprise Tech-Stack

Für die Realisierung dieser hochentwickelten RAG-Architekturen im Rahmen unserer Kundenprojekte (u. a. für die KI-gestützte Suche bei *KW Voerde* sowie die interne Slack-Bot-Architektur von *Solarize*) hat sich in der rms.-Entwicklungspraxis folgender System-Stack bewährt:

Pipeline-Phase	Technologie (rms.-Standard)	Funktionale Begründung im Enterprise-Einsatz
Parsing & Layout Extraction	**Kreuzberg-PDF / Tesseract / PaddleOCR**	Optimierter Hybrid-Ansatz für strukturierte PDFs, gespannte Bounding-Boxes und hochpräzise zweidimensionale Tabellenextraktion.
Vector Database & Indexing	**ChromaDB**	Skalierbare, hochperformante Speicherung dichter Vektoren mit nahtloser Integration in agentische Workflows und Metadaten-Subspace-Filtering.
Orchestration Layer	**LangChain / LangGraph / Neuron AI**	Zustandsgesteuerte Multi-Agenten-Systeme (LangGraph) und robuste LLM-Abstraktionen (LangChain) gepaart mit spezialisierten logischen Kernels (Neuron AI).
Reranking Engine	**Jina / Zeroentropy / IONOS (Qwen3-VL Reranker-8B)**	Multimodales und tiefes Cross-Encoder-Reranking zur Filterung mathematischen Rauschens vor der LLM-Injektion.

5. Fazit und Ausblick

Die zuverlässige Verarbeitung geschäftskritischer Dokumente erfordert den konsequenten Abschied von naiven Text-Chunking-Routinen. Wenn geschäftskritische Daten in Tabellen, Grafiken und Belegen fehlerfrei zugänglich gemacht werden sollen, müssen Layout-Awareness, multimodale Indexierung und hybride Filterverfahren softwareseitig fest verankert werden. Durch den Einsatz des hier beschriebenen Enterprise-Setups transformiert rms. unstrukturierte, historisch gewachsene PDF-Bestände in mathematisch präzise, auditierbare und hochperformante Wissensnetzwerke.

rms. relationship marketing solutions GmbH
Calwer Straße 23 | 70173 Stuttgart | Geschäftsführer: Oliver Mack
Web: rm-solutions.de | IT & Entwicklung: Michael Kettel (Head of IT)