RAG System Methods, Tools, and Technologies - Clean Comparison

---

1. Data Extraction

Method / Tool	Description	Supported Formats
PyPDF2 / pdfplumber	Python libraries for extracting text from PDFs	PDF
Unstructured.io	Extracts and cleans text from multiple document types	PDF, DOCX, PPTX, HTML, images
LangChain Document Loaders	Built-in loaders for PDFs, text files, web pages, and databases	All major formats
LlamaIndex Data Loaders	Connectors for structured/unstructured sources	PDFs, SQL, Notion, Google Drive
BeautifulSoup / Scrapy	Extracts text from web pages	HTML, XML
Textract / Tesseract OCR	Extracts text from scanned images and PDFs	Images, scanned docs
API-based Extraction	Retrieves data from APIs or databases	JSON, CSV, DB tables

---

2. Data Chunking

Method / Tool	Description	Chunking Approach
RecursiveCharacterTextSplitter	Splits text recursively by paragraphs, sentences, and characters	Recursive splitting
TokenTextSplitter	Splits based on token count	Token-based
SentenceSplitter	Splits text into sentences	Sentence-based
Unstructured.io partitioning	Auto-splits documents during extraction	Hybrid

Method / Tool	Description	Chunking Approach
Custom Regex Splitter	Splits using patterns like section headers	Rule-based
Semantic Chunking	Splits based on semantic similarity	Meaning-based

3. Embeddings

Embedding Method / Model	Developer	Dimension Size
OpenAI Embeddings	OpenAI	1536 / 3072
HuggingFace Sentence Transformers	HuggingFace	384 / 768
Instructor Embeddings	HKU NLP	768 / 1024
Cohere Embeddings	Cohere	4096
Google Vertex / Gemini	Google Cloud	Varies
Local Transformer Models	Open-source	768–1024

4. Vector Databases

Vector DB	Type	Storage Mode
FAISS	Local / Open-source	In-memory / disk
ChromaDB	Local + Cloud	In-memory / persistent
Pinecone	Cloud-based	Managed cloud storage
Weaviate	Cloud + Local	Persistent
Milvus	Cloud + Local	Distributed
Qdrant	Cloud + Local	Persistent
Elastic Vector Search	Cloud + Local	Integrated into Elasticsearch

5. Retriever

Method	Technology / Tool	How it Works
Similarity-based Retriever	FAISS, Chroma, Pinecone	Uses cosine similarity or dot product
BM25 / Keyword Retriever	ElasticSearch, Whoosh	Lexical matching
Hybrid Retriever	LlamaIndex, LangChain	Combines semantic + lexical retrieval

Method	Technology / Tool	How it Works
Multi-vector Retriever	Milvus, Weaviate	Multiple embeddings per document

6. Generator (LLM Integration)

Model Type	Examples	Strengths
OpenAI Models	GPT-4, GPT-3.5	Accurate, handles context well
Open-source LLMs	Llama 3, Mistral, Falcon	Customizable offline models
Instruction-tuned Models	Vicuna, Alpaca	Follows prompts well
Domain-Specific Models	BioGPT, LegalBERT	Specialized in healthcare/legal

7. Prompt Engineering

Type	Example	Purpose
Zero-shot Prompting	“Answer based on context below.”	Simple Q&A
Few-shot Prompting	“Input… Output…”	Guides LLM with examples
Chain-of-thought Prompting	“Explain reasoning step-by-step.”	Encourages reasoning
Contextual Prompting	Add retrieved documents	Combines context + query
Instruction-tuning Prompting	“You are an expert…”	Domain control

8. Memory

Type	Tool / Implementation	Function
Buffer Memory	ConversationBufferMemory	Stores complete chat history
Summary Memory	ConversationSummaryMemory	Stores summarized context
Vector Memory	Chroma, Pinecone	Semantic memory retrieval
Combined Memory	Hybrid	Advanced chatbots

9. Evaluation & Deployment

Tool / Platform	Use / Purpose	Description
RAGAS	Evaluate relevance, faithfulness, context precision	Metrics for RAG performance
TruLens	Model explainability	Tracks reasoning and performance
PromptLayer	Prompt versioning & monitoring	Tracks LLM behavior
Weights & Biases	Experiment tracking	Logging, visualization
Gradio / Streamlit	UI-based deployment	Chatbot interface
FastAPI / Flask	API-based deployment	REST API endpoints
AWS / GCP / Azure	Cloud hosting	Scalable enterprise deployment
Docker + Kubernetes	Containerization	Portable, cloud-native apps