Haystack:

Haystack is an open-source framework by deepset for building search systems, QA systems and RAG applications with the integration of LLMS along with the ready-to-use application. It focusses on the DAG based systems (Directed Acyclic Graph) that can be used for the building of the real-world RAG / LLM pipeline.

An end-to-end open-source RAG / AI-orchestration framework focused on document ingestion, indexing, retrieval, and LLM-backed QA and agentic pipelines. Enterprise-grade features, vector DB integrations, evaluation tooling.

Goal:

Robust, production RAG: index many docs, retrieve accurately, serve LLM answers grounded in documents (search/QA).

Contents:

✓ Why Haystack introduced
✓ Core components
✓ Architecture
✓ Strength of Haystack
✓ Weakness of Haystack
✓ Why Haystack for Medical Systems
✓ Tools Haystack Integrates
✓ When Should You Use Haystack?
✓ Evaluator

Why Haystack:

Traditional LLM applications are faced with hallucination and content mismatch as LLM are trained on the basic data that is provided at that time. To avoid the content mismatch or hallucination, RAG systems are introduced.

Haystack gives you:

- Document ingestion
- Preprocessing
- Embedding
- Vector database indexing
- Retrieval
- Answer generation
- Pipelines

- Evaluator

All with production-level tools.

Core Concepts:

1. Document Store: where documents stored + with embeddings

2. Nodes: Processing units in the pipeline

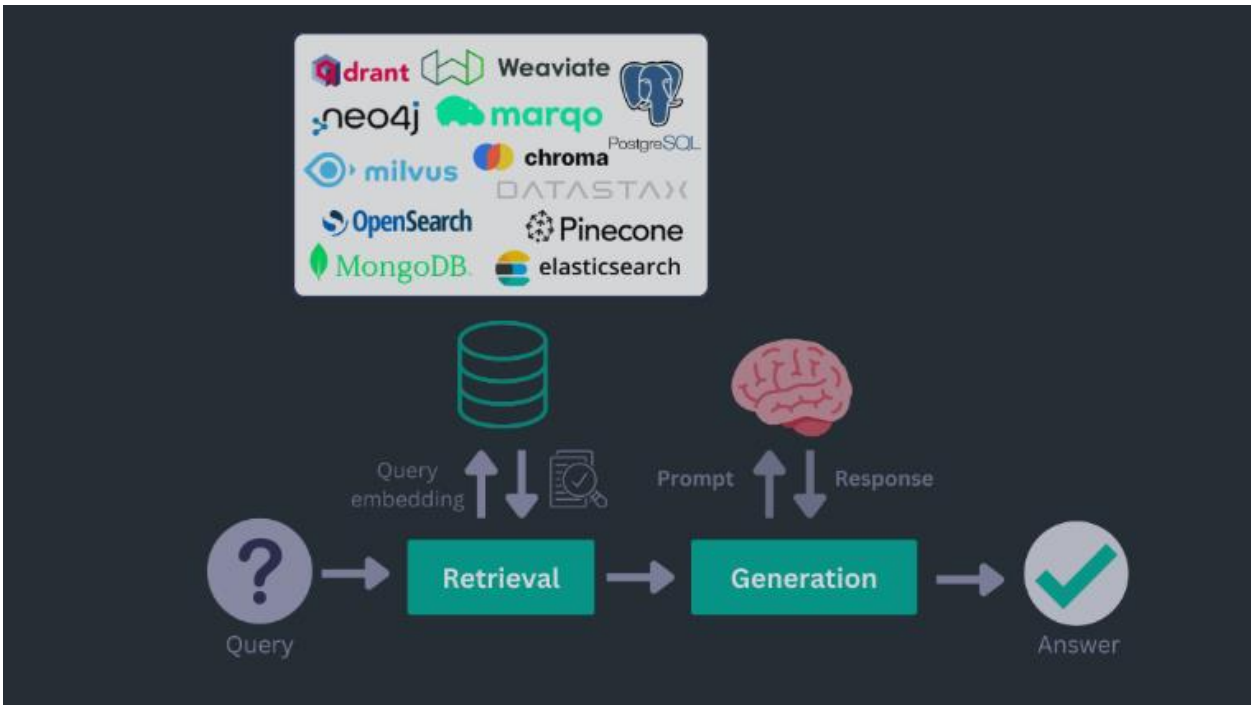
3. Pipeline: DAG style pipeline

4. Retrievers: Retrieves the content embeddings

5. Generators: Operational Model that generates the content with the retrieved contents

6. Agents: Agentic LLMs that has access to the functional tools

Architecture:



Strength of Haystack:

✓ Best open-source RAG engine

Highly accurate document retrieval.

✓ Production ready

Used by enterprises (BMW, Airbus, Intel).

✓ Modular & flexible

You can customize every component.

✓ Great for large datasets

Millions of documents.

✓ Strong for healthcare, legal, research

Due to accurate retrieval + audit workflows.

✓ Multi Document fomat support

You can ingest almost any file (PDF, DOCX, PPTX, TXT, HTML,JSON Excel)

Weakness of Haystack:

❌ Not as "agent-friendly" as LangChain (but agents exist)

❌ More backend-focused

❌ Steeper learning curve for beginners

❌ Fewer tutorials compared to LangChain

Why Haystack for Medical Systems

The Basic need of the Medical Chat system:

1. Medical Rag systems
2. High-accuracy retrieval
3. Multi-step pipelines
4. Explainable reasoning
5. Control + reproducibility
6. EMR / guidelines integration
7. Large document ingestion
8. Low hallucination

Haystack Excels:

- High Accurate retrieval of contents
- Large Document Handling
- multiple formats and both structure and Unstructure DBs like (FAISS Weaviate Milvus Qdrant Elasticsearch Pinecone OpenSearch PostgreSQL SQLite)
- Medium learning curve compared to other framsworks like langGraph
- Ideal choice for the Medical and Legal and research purposes

Tools Haystack Integrates:

- Vector DBs
- LLM providers (OpenAI Hugging face, Google, Anthropic, Local Models with Ollama)
- Deployment Applications (Docker, Kubernetus REST API server)

When Should You Use Haystack?

Pick *Haystack* if your priority is **reliable, high-quality Retrieval-Augmented Generation (RAG)**, document indexing, semantic search and production readiness for large medical corpora.

Use Haystack if:

✓You need accurate RAG

✓ You're building a medical/enterprise search system

✓ You need to index large document sets

✓ You want structured pipelines

✓ You require explainability and production stability

Do NOT choose Haystack if:

- You only need simple prompting
- You want heavy agent workflows (LangGraph may be better)

Why Evaluator Exists — Purpose

Haystack is built for production-grade RAG, which means you must measure performance before and after changes.

The Evaluator ensures:

- Your system retrieves the right evidence
- Your LLM produces faithful responses
- Your changes (new embeddings, new prompts, new dataset) do NOT silently degrade accuracy
- You can compare pipelines using objective metrics

For medical AI, this is crucial to avoid hallucination and ensure patient-safe answers.

Types of Evaluators:

1. Retriever Evaluator
2. Generator Evaluator

3. Full Pipeline Evaluator

Retriever Evaluator

Checks whether the retriever returns the correct documents for a question.

Metrics include:

Recall@k
→ Did the correct document appear in the top-k results?

Mean Reciprocal Rank (MRR)
→ How early in the ranked list does the right document show up?

Precision@k

MAP (Mean Average Precision)

F1 score

Use when comparing:

- BM25 vs Dense Retriever
- Different embedding models
- Different vector DB configurations
- Hybrid vs dense only

Reader / Generator Evaluator

Evaluates whether the LLM's answer matches the ground-truth answer.

Metrics include:

Exact Match (EM)
→ Strict string match (mainly for extractive QA)

F1 score (token-level)

Answer Similarity (semantic)

Faithfulness (alignment with retrieved context)

Context Utilization
→ Checks how much of the retrieved documents were used.

Useful for RAG LLMs because EM is too strict; semantic and faithfulness scores matter more.

Full Pipeline Evaluator (End-to-End)

Evaluates the entire RAG system:

retriever → reranker → LLM → output

Metrics:

Retrieval Recall@k

Reader/LLM F1

Pipeline EM

Faithfulness (does the answer cite/evidence from retrieved docs?)

Hallucination Rate

Latency

Useful when:

You add new sources (e.g., iCliniq, government sites)

You modify prompts

You fine-tune a new LLM

You change indexing or embeddings

How Evaluator Works:

1. You prepare a dataset (e.g., JSON, SQuAD format).
2. You run the pipeline (retriever + generator) on each query.
3. The Evaluator compares output vs expected values.
4. Metrics and logs are produced.
5. You visualize and compare results across pipeline versions.

Why Evaluator is Critical for Medical AI:

Because medical applications are high-risk, evaluation must be:

- consistent

- automated
- clinical grade

Evaluator helps guarantee:

- The query returns the correct medical guideline
- The LLM does not hallucinate beyond retrieved evidence
- Reliability improves over time