



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



Discuss →

Solution →

Day 1: Fetching Holiday Wisdom for Santa

With Christmas looming, the North Pole was buzzing with activity. Toys were being assembled, reindeer were training for the big night, and the elves were fine-tuning Santa's sleigh. But amidst the holiday chaos, Santa faced a pressing personal challenge.

For the past year, Santa had been learning all about LLMs, RAG, and AI Agents and loving it. These approaches had revolutionized his work last Christmas, and he wanted to take things even further this year.

“I need to finish these blog posts and understand how to transform queries for better retrieval and the difference between metadata filtering and metadata enrichment,” Santa muttered, pacing his study. “But Christmas is almost here, and there’s no time before I’m buried in gift deliveries and chimney logistics!” 

“Santa, I overheard you,” said Elf Bilge, popping into the room. “You want to master advanced RAG approaches before Christmas, but there’s no time to read everything. What if we build something that can fetch the articles you want and answer your questions directly?”

Santa’s eyes lit up. “Why didn’t I think of that? You’re right, Bilge. I guess I’m getting old. We just need a Haystack pipeline with a **LinkContentFetcher** and a few extra components, and we’re all set!” 

Elf Bilge grinned. “With a little Haystack magic, anything’s possible.”

For this challenge, you must help Elf Bilge build a Haystack pipeline that fetches content from given URLs, processes the data for relevance, and enables a seamless Q&A system to answer Santa’s queries

Here are the initial components you will use for this challenge:

- **LinkContentFetcher** for using the contents of several URLs in your pipeline
- **HTMLToDocument** for converting the HTML files into documents.
- **PromptBuilder** for creating the prompt
- **OpenAIGenerator** for generating responses

Your task is to figure out how to connect these components and think about additional components to add this pipeline that will help you identify the 10 most relevant chunks of the given content.

Requirements:

- An **OpenAI API Key** if you’d like to use **OpenAIGenerator** but you can choose any other LLM that is supported with **Haystack Generators**



Some Hints:

- Check out the [Haystack Pipelines documentation](#) to learn more about pipeline connections
- If you connect the given components, the initial pipeline will work as is, but there are two additional components you can add to enhance this pipeline.
- No retriever is necessary for this challenge
- One extra component is to split the documents into smaller chunks of around 10 sentences each.
- As the other component, consider methods or components you can use to filter out irrelevant chunks by ranking before injecting the context into the prompt.

★ **Bonus Task:** Use the OpenAIChatGenerator or other ChatGenerator component instead of the standard Generator component and provide a system message to guide the LLM:

```
system_message = """You are a technical expert. Use only the provided documents to answer the question. Do not add any extra information or search the web. Keep your answers clear, accurate, and to the point, including the document source. If the documents don't have the answer, say "no answer""""
"""
```

♥ Here is the [Starter Colab](#). Don't forget to submit your solution notebook using the “Submit” button above to win surprise prizes!



Building products, technology
and solutions for LLM-enabled



COMMUNITY

[GitHub Discussions](#)



Advent of Haystack solutions are here, explore them now! →

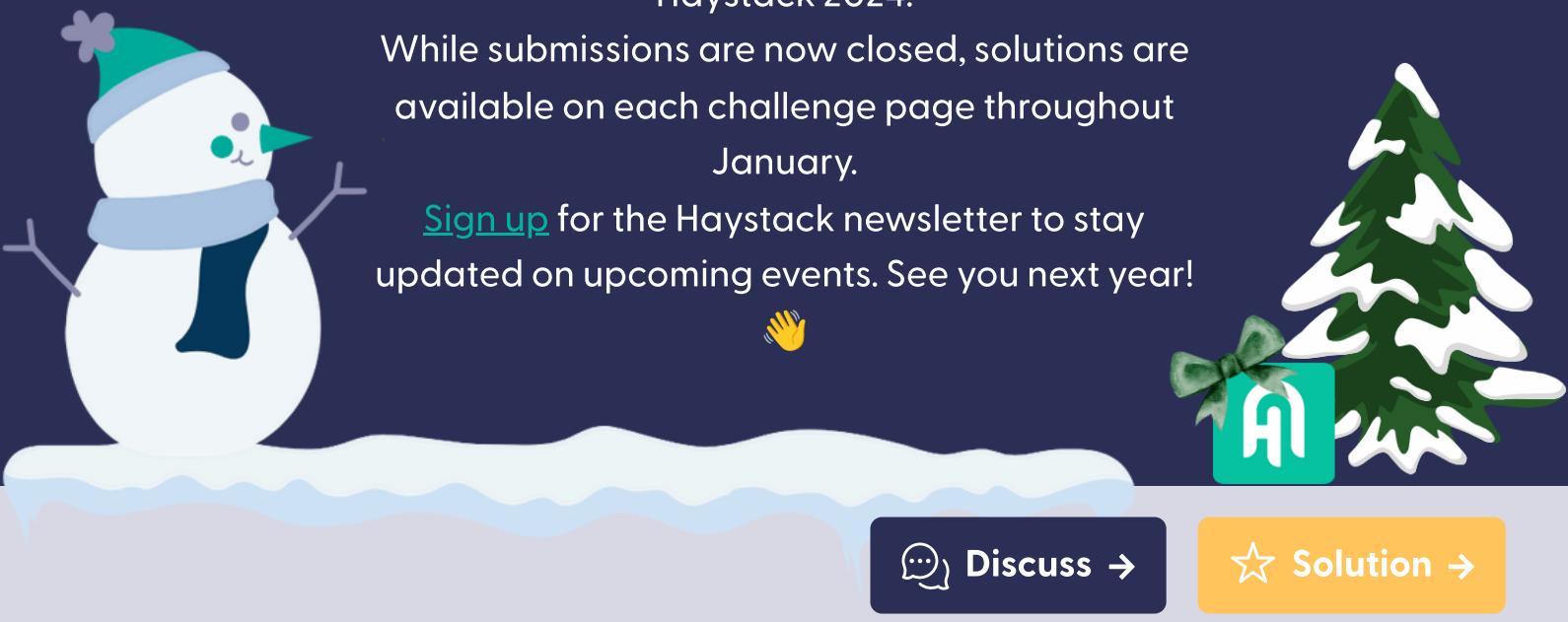
Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎅

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



Discuss →

Solution →

🎄 Day 2: Help the Elves Find What They've Lost



This challenge will have you break the fourth wall of a holiday film, where for a few minutes, you'll be walking a line between fiction and reality.

🔍 Challenge Objective:

For today's challenge, we're going to be visiting the fictional universe of the Weaviate Elves in the film "A Very Weaviate Christmas". There seems to be some trouble brewing. Something has been stolen from them. Your challenge is to find out what's going on and who is behind it.

You will know that you've successfully completed the challenge if you can answer the question "Who is the culprit?"

🎄 Some things to get you started

In the starter Colab you will find:

- A `WeaviateDocumentStore` which has access to information about many many movies
- A (read-only) Weaviate API Key for you to be able to access the Collection we've provided
- The URL for the Weaviate Client to use.
- Lots of hints to help you along the way
- Links to documentation that will be useful in your quest!

🎯 Requirements

💡 Here's the Starter Colab

📚 Useful Documentation and Resources

- [Tutorial: Creating Your First QA Pipeline with Retrieval-Augmentation](#)
- [OpenAITextEmbedder](#)
- [OpenAIGenerator](#)
- [PromptBuilder](#)
- [WeaviateDocumentStore](#)
- [WeaviateEmbeddingRetriever](#)
- [Weaviate Documentation: Read all Objects](#)
- [Weaviate Documentation: Filters](#)



Building products, technology
and solutions for LLM-enabled
applications.



COMMUNITY

[GitHub Discussions](#)

[Discord](#)

[Hugging Face](#)

[Open NLP Meetup](#)

RESOURCES

[Models](#)

COMPANY

[About](#)



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎅

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



[Discuss →](#)

[Solution →](#)



Day 3: Making Answers Bright with Multi-Query RAG Magic ✨

Santa Claus realized his centuries-old tradition of spreading joy was missing something in the modern age—knowledge about the world beyond his snowy workshop. While he knew about children’s hopes and dreams from their letters, Santa felt increasingly out of touch with the broader events shaping their lives.

“How can I be a true global gift-giver,” Santa wondered, “if I don’t understand what’s happening in the world?”

He called an emergency meeting with tech-savvy Elf David, who had been itching for a project beyond sleigh upgrades. Together, they decided to build a Retrieval-Augmented Generation (RAG) system using Haystack, which Elf David had previously used to streamline toy inventory tracking. This time, they connected it to the BBC News feed to create a personal RAG assistant for Santa.

“One last thing,” Santa added, “I know about advanced retrieval methods. Use multi-query retrieval to improve recall—I want the most relevant answers!”

For this challenge, help Elf David create custom components for multi-query retrieval in the RAG pipeline.

Requirements:

- An [OpenAI API Key](#) if you'd like to use [OpenAIGenerator](#) but you can choose any other LLM that is supported with [Generators](#)



Some Hints:

- Check out the [Creating Custom Components](#) to learn more about how to create custom Haystack components
- Some blog post links we used on Day 1 can be helpful 😊



Here's the [Starter Colab](#)



Building products, technology
and solutions for LLM-enabled
applications.



[COMMUNITY](#)

[GitHub Discussions](#)



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



[Discuss →](#)

[Solution →](#)

Day 4: Sharing the Christmas Magic with Gen Z 🎵

At the North Pole, where magic and technology danced hand in hand, Santa was preparing for Christmas. In his workshop filled with glowing gadgets and humming machines, he summoned Elf Patrick, one of the most tech-savvy elves. “Elf Patrick,” Santa said with a chuckle, “this year, I want you to share the story of how I became Santa Claus. But here’s the twist—you’ll use coding to help you create the perfect versions of the story for all kinds of kids.”

Elf Patrick’s eyes sparkled. “Coding? That’s my jam, Santa! I’ll make sure your story is ready for everyone, no matter how they like it told!” Santa handed Elf Patrick the

audio file. “Start by transcribing it, then simplify it, and finally, remix it into that fun Gen Z lingo I keep hearing about.” “Consider it done!” Elf Patrick declared, racing to the North Pole Coding Hub.

For this challenge, you must help Elf Patrick transcribe Santa’s story, summarize it, and rewrite it for Gen Z kids.

Here are the components you might need for this challenge:

- [AssemblyAI Transcriber](#) for speech-to-text tasks
- [PromptBuilder](#) for creating the prompt
- [OpenAI Generator](#) for generating responses

Requirements:

- A free [AssemblyAI API key](#)
- An LLM. Here we’ll use an [OpenAI API key](#), but you can choose any other LLM that is supported with [Haystack Generators](#)

Some Hints:

- [Haystack docs - AssemblyAI integration](#)
- [Haystack docs - Creating pipelines](#)

 Here is the [Starter Colab](#)





Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



[Discuss →](#)

[Solution →](#)

Day 5: Elves' Secret for Faster Development



In the buzzing Tech Hub of the North Pole, Elf Tobi was at work, surrounded by glowing monitors and mugs of hot cocoa. Known for building quick and modular solutions, Elf Tobi thrived on AI challenges. Because he had a secret 😊

Elf Tobi was a Haystack fan and had recently started using **deepset Studio** to make the pipeline-building process even more fun and efficient. Its drag-and-drop interface made building and editing Haystack pipelines a breeze, turning a challenging task into a fun, interactive experience. He especially appreciated the deployment and testing features, as well as the handy pipeline templates that

provided a great starting point. “Deploying and testing on the Playground? That was like adding a shiny star on top of the Christmas tree!” Elf Tobi thought about deepset Studio 🎄

For this challenge, you’ll join Elf Tobi to create a RAG pipeline in deepset Studio. Here are the steps to complete the challenge:

- Use sample files for “AI” in deepset studio to populate the Elves’ AI Library
- Explore pipeline templates and use the “RAG Question Answering GPT-4o” template (or any other template) to save time 😊
- (Optional) Replace the models in the pipeline, use different providers or open source ones.
- Build, save, and deploy your indexing and query pipeline (Deployment might take a bit of time, so feel free to revisit earlier Advent of Haystack challenges while you wait 🎅)
- Use the Playground to test your pipeline and submit the pipeline’s output for the query “What’s Compound AI?”

Helpful Links:

- Learn more about deepset Studio features in [our announcement post](#)
- If you have any questions, feel free to ask in the “#deepset-studio” channel on [Discord](#)

Some Hints:

- Follow the onboarding flow of deepset Studio to learn how to create, deploy, and test your pipeline on the platform.
- Watch the recording of [Fast-Track Your LLM Apps with deepset & Haystack](#) for a complete walkthrough from Elf Tobi.
- Learn how to add API keys, edit and deploy pipelines or generate a shareable pipeline link in the [deepset studio documentation](#)

Bonus Task

- Extend your pipeline's capabilities and build a RAG pipeline with a fallback to the web. This updated pipeline should query the Elves' AI Library first, and if no answer can be found, fetch it from the web.
- As the fallback, you can either just search for the results on the Web or use these Web results to generate the complete answer.
- Get your free API key from [SearchAPI](#) or [Serper](#) and [add the key to your connections/secrets](#) to use web search components in your pipeline.
- Get inspiration from [Tutorial: Building Fallbacks to Websearch with Conditional Routing](#) to understand how to design the pipeline with a fallback mechanism.

 **GET STARTED:** Sign up for [deepset Studio](#) (free and open to everyone) and get access within 24 hours



Building products, technology
and solutions for LLM-enabled
applications.



COMMUNITY

[GitHub Discussions](#)
[Discord](#)
[Hugging Face](#)
[Open NLP Meetup](#)

RESOURCES

[Models](#)
[Datasets](#)

COMPANY

[About](#)
[Jobs](#)



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



[Discuss →](#)

[Solution →](#)

Day 6: Keep the Holiday Spirit Alive



There's always so much to do to prepare for Christmas! Toys to build, visitors to greet, and machines to maintain... it's a whirlwind in Santa's workshop. Luckily, Santa has mastered the art of delegation over the years. This time, he's entrusted the ever-resourceful Elf Rita to ensure everything runs like clockwork! ⏳

Elf Rita is managing a growing team of elves, each with unique skills, and an ever-expanding list of holiday tasks. As one of the tech-savvy elves, Elf Rita turns to her favorite tool: **NVIDIA NIM microservices**, confident it's the key to tackling her tasks

efficiently. She eagerly dives into her first challenge, **Optimal Elf Assignment**, dreaming of a well-deserved break before the big day.

But just as she's about to relax and sip some hot cocoa, Santa surprises her with another "rewarding" job: **Delivery Organiser**. He promises it's the most critical task yet!

Now, Elf Rita needs your help to complete her assignments using **NIM microservices** hosted on the NVIDIA API catalog. Can you step in to save Christmas and keep Elf Rita's holiday spirit alive? 🎄

For this challenge, there are two tasks to tackle:

- Task 1: Optimal Elf Assignment
- Task 2: Delivery Organiser

🎯 Requirements:

- A free API key with 1k credits to be able to use models hosted via the [NVIDIA API catalog](#).

💡 Some Hints

- Check out the [Article: Design Haystack AI Applications Visually in deepset Studio with NVIDIA NIM](#) or [Building RAG Applications with NVIDIA NIM and Haystack on K8s](#) to learn more about NVIDIA NIMs and how to use them with Haystack
- You will only need these three components: [NvidiaRanker](#), [NvidiaTextEmbedder](#), [DocumentMRREvaluator](#)
- More tips in the Starter Colab [!\[\]\(5e9f39cb78cebf673264ace742bd09e8_img.jpg\)](#)

💚 Here's the Starter Colab



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



[Discuss →](#)

[Solution →](#)

Day 7: Judging Toys, Tracing Joy





Santa collapsed into his chair with a huff, settling heavily next to Mrs. Claus.

🎅: "What's wrong?"

🎅: "There's just too many toys to check and not enough time! Christmas is almost here!"

🎅: "Well, can't you just check some of them?"

🎅: "I wish it were that easy! But my elves make so many different toys, and we have to ensure every kid gets the right one!"

Elf Jane overheard the conversation from the next room. As a regular attendee at the North Pole Hackathon, she had learned a lot about evaluation recently and thought she might have a solution. "What if I build an **LLM Judge** to help?" she thought. "I can use Arize Phoenix to log everything—like why this toy was the perfect match or why it wasn't!"

For this challenge, you will help Elf Jane by:

- Using a **Haystack pipeline** to find the best toy for each child in the Big Elf Database of Christmas Wishlists (BEDCW)
- Evaluating all toy matches using an **LLM-as-a-Judge**

- Monitoring the system with the open-source tracing and evaluation tool, [Arize Phoenix](#).

Requirements:

- An [Open API Key](#) if you'd like to use [OpenAIChatGenerator](#) but you can choose any other LLM that is supported with [Haystack Generators](#)

Some Hints

- Take a look at this example notebook: [Tracing and Evaluating a Haystack Application with Phoenix](#)
- Find more examples in [Arize Phoenix Docs](#)

 Here's the Starter Colab



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



Discuss →

Solution →

Day 8: Ultimate Inventory Agent to Control Holiday Chaos 🔧🎄



The North Pole workshop was buzzing with activity as elves rushed to assemble toys and wrap gifts. Amid the chaos, Elf Stefano, known for his love of language models and impeccable inventory management, found himself overwhelmed. 😞

His usual system of sticky notes and spreadsheets was falling apart. “I can’t keep track of what’s in stock, what’s gone, or if we’re out of googly eyes for teddy bears!” he groaned, staring at his cluttered desk.

Elf Stefano wasn’t just any inventory manager; he was the workshop’s language model enthusiast. He had trained models to craft personalized replies to children’s letters and help Santa write poetic thank-you notes. But this year, his passion for language models had taken a backseat to managing endless inventory demands.

To make matters worse, elves bombarded him with questions: “Elf Stefano, do we have enough paint for toy trains?”, “What’s the price of candy canes on Amazon?”, “We’re out of jingle bells again!” 😳

Elf Stefano needed an innovative solution to handle the chaos. That’s when inspiration struck; he would build an 💡🤖 INVENTORY AGENT 🤖💡 using Haystack! The Agent would manage inventory, track items, and fetch prices while Elf Stefano returned to fine-tuning magical language models to spread holiday cheer 🎅

For this challenge, join Elf Stefano and help complete the three missing tools for the ultimate inventory agent. For this challenge, you will use some of the [haystack-experimental](#) components:

-  [OpenAIChatGenerator](#)
-  [ToolInvoker](#)

Requirements:

- An [OpenAI API Key](#) if you'd like to use [OpenAIGenerator](#) but you can choose any other LLM that is supported with [Haystack Generators](#)

Some Hints

- To learn about Tools and Agent implementation, check out [Cookbook: Define & Run Tools](#) and [Cookbook: Newsletter Sending Agent](#)
- Learn more about [haystack-experimental](#) in [Docs: Experimental Package](#)

 Here's the [Starter Colab](#)



Building products, technology
and solutions for LLM-enabled
applications.



COMMUNITY

- [GitHub Discussions](#)
- [Discord](#)
- [Hugging Face](#)
- [Open NLP Meetup](#)



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



Discuss →

Solution →

Day 9: Santa's Self-Reflecting Gift Agent





Elf Pash at the North Pole has a challenge. Every year, Santa needs to process millions of gift requests and optimize his gift selections based on children's wishlists and budget constraints. Other elves often ask questions like "What gifts should we get for a 9-year-old who loves science?" or "How can we maximize joy while staying within budget?" 🎁

This year, elf Pash has an idea to solve this: create a self-reflecting agent that can optimize gift selections automatically! As he experiments with LLMs, he realizes that simple gift suggestions aren't enough - the agent needs to reflect on its choices to ensure they're optimal. Being a Haystack elf, Pash knows how to solve this: SELF-REFLECTION! 🌙

So, he comes up with a plan. Santa will create a gift recommendation system using Haystack's RAG pipeline with MongoDB Atlas vector search, enhanced with a self-reflecting component that optimizes gift selections based on budget, age appropriateness, and joy factor! ✨

For this challenge, you must help elf Pash create a pipeline that can suggest and optimize gift selections through self-reflection.

- Implement a self-reflecting agent that can optimize gift selections

- Use MongoDB Atlas vector search for semantic gift matching
- Include price, age range, and category in gift considerations
- Ensure all suggestions stay within the specified budget

Requirements:

- To use the default embedding and text generation models in this notebook, you'll need an [OpenAI API Key](#). Alternatively, you can replace these models with others that better suit your needs.
- [MongoDB Atlas project](#) with an Atlas cluster (free tier works). Create your cluster, take a note of your [connection string](#) and have `0.0.0.0/0` address in your network access list. Visit [detailed tutorial](#) for step by step guide.

Some Hints:

- Include gift metadata in embeddings with `meta_fields_to_embed` of your document embedder
- Create a custom component for gift optimization checks, you have seen this in day 3
- Use the `max_runs_per_component` parameter in Pipeline for controlled self-reflection
- Enable [Real-Time Pipeline Logging](#) to inspect the data that's flowing through your agentic pipeline
- Check our DeepLearning.AI course [Building AI Applications with Haystack](#) or [Web-Enhanced Self-Reflecting Agent](#) for similar examples

 Here is the [Starter Colab](#)



Advent of Haystack solutions are here, explore them now! →

Advent of Haystack

Explore Haystack with Weaviate, AssemblyAI, NVIDIA, Arize AI, and MongoDB through 10 challenges! 🎉

♥ Thank you for your interest in the Advent of Haystack 2024.

While submissions are now closed, solutions are available on each challenge page throughout January.

[Sign up](#) for the Haystack newsletter to stay updated on upcoming events. See you next year!



Discuss →

Solution →

Day 10: Jingle Metrics All the Way



Haystack Elves worked tirelessly this year to make the holiday season stress-free and joyful. Determined to innovate, they tackled challenges with cutting-edge AI solutions.

They enhanced pipelines with speech-to-text models, explored various LLM providers, and customized Haystack pipelines for unique needs. They built AI Agents with tool-calling and self-reflection, added tracing mechanisms, and developed faster with deepset Studio. To ensure a top-notch tech stack, they partnered with tools like [Weaviate](#), [AssemblyAI](#), [NVIDIA NIMs](#), [Arize Phoenix](#), and [MongoDB](#).

However, there's one crucial step remaining before taking anything into production:

Evaluation

Haystack equips the elves with the tools they need, including [integrations with evaluation frameworks](#) and built-in [evaluators](#). Adding to this, the Haystack ecosystem now features a powerful new tool: [EvaluationHarness](#). This tool streamlines the evaluation process for Haystack pipelines by eliminating the need to create a separate evaluation pipeline while also making it easier to compare configurations using overrides.

For this challenge, you need to help Haystack elves evaluate a simple RAG pipeline using [RAGEvaluationHarness](#), a specialized extension of [EvaluationHarness](#) designed to simplify and optimize evaluation specifically for RAG pipelines.



- A [Hugging Face API Key](#) with access to free gated models, [meta-llama/Llama-3.2-1B-Instruct](#) and [meta-llama/Llama-3.2-3B-Instruct](#). Visit the model pages to request access. More details are in the Starter Colab below.
- An [OpenAI API Key](#) to use LLM-based evaluators with `EvaluationHarness`, such as [FaithfulnessEvaluator](#), [ContextRelevanceEvaluator](#)

💡 Some Hints:

- Explore the [Walkthrough: Evaluation](#) to learn all about evaluation in Haystack.
- For practical examples, check out [Cookbook: Evaluating RAG Pipelines with EvaluationHarness](#) and [Cookbook: Evaluating AI with Haystack](#).

⭐ **Bonus Task:** Take it a step further by incorporating **hybrid retrieval** into your pipeline. Use `EvaluationHarness` with customizations to test whether hybrid retrieval improves Recall and MRR 🤔

❤️ Here is the [Starter Colab](#)



Building products, technology
and solutions for LLM-enabled
applications.



COMMUNITY

[GitHub Discussions](#)
[Discord](#)
[Hugging Face](#)