Additional content has been loaded

SKIP TO LESSON

EXIT COURSE

**Lesson 1 of 3**

[**Generative AI Solutions - Evaluation of RAG Systems**](https://uhg.edcast.com/lxphubservice/content/private/300439/cninv-21856984-2db8-4b00-97b1-d56e5dbf80c0/scormcontent/index.html#/)

[**Home**](https://uhg.edcast.com/lxphubservice/content/private/300439/cninv-21856984-2db8-4b00-97b1-d56e5dbf80c0/scormcontent/index.html#/)

**Evaluating the Retriever Component**

**Lesson 1 of 3**

In the field of information retrieval—particularly when working with large language models, semantic search, or chunked document retrieval—it’s not enough to simply fetch results. What truly matters is how relevant and complete those results are in the context of the user’s query. This becomes even more critical in the healthcare industry, where accuracy and comprehensiveness can directly influence clinical decisions, patient safety, and the quality of research.

For example, when retrieving documents to support diagnosis of a rare disease, it’s not sufficient to return articles that merely mention the symptoms. The retrieved content must accurately reflect the clinical context described in the query.

Traditional metrics like precision and recall provide a foundational framework for evaluating performance. However, when the goal is to assess semantic alignment—that is, whether the retrieved documents match the meaning and intent behind a query—more refined measurements are needed.

This is especially important when working with generative AI systems such as retrieval-augmented generation (RAG), where relevance goes beyond keyword matching. In these systems, it’s essential that the underlying meaning of the retrieved content supports or informs the generated response.

This section introduces three essential concepts that help evaluate retrieval quality in modern systems:

1. **Contextual Precision / Precision@K:** How accurate are the top K retrieved results vs. all retrieved results in terms of contextual relevance?
2. **Contextual Recall / Recall@K :**How many of the truly relevant results are we retrieving in the top K results vs. from the entire dataset?
3. **Contextual Relevancy:**How well do the retrieved chunks semantically align with the intent of the query?

Whether you're building a medical search engine, developing an LLM for clinical Q&A, or designing a symptom-based triage RAG system, understanding these concepts is key to delivering the right results.

**Contextual Precision**

Let’s begin with contextual precision. This metric evaluates all the chunks the system retrieved for a given query—whether that’s 10, 20, or more. It helps answer the question:  
**“Out of everything the system returned, how much was actually useful and relevant to what the user asked?”**

This type of precision goes beyond simple keyword matching. It also considers whether each chunk makes sense in the context of the query’s intent.

Contextual precision is computed using the formula:

This formula tells us how accurate the retrieval was overall. It looks at all the chunks the system retrieved and checks what percentage of them were relevant to the user's query.

Query

Example

Result

Why this matters

What are the challenges of using wearable blood pressure monitors in home-based hypertension management?

**Precision@K**

Now, let’s look at **precision with a cutoff**, commonly referred to as Precision@K. This metric only looks at the  
**top K** chunks and asks:  
**“How many of the top K results were actually useful in the context of the query?”**

This matters because, in real-world applications, users—especially clinicians, technicians, or healthcare analysts—rarely scroll through every result. They often rely on the top few responses to make quick decisions or form initial judgments.

Precision@K is computed using the formula:

Query

Example

Result

Why this matters

What are the challenges of using wearable blood pressure monitors in home-based hypertension management?

It’s also important to note that Precision@K isn’t just about whether relevant results appear somewhere in the list—it’s also about where they appear. Most retrieval systems return results in ranked order based on their scoring models. If the relevant chunks appear at the top, that’s ideal. However, if they’re buried further down, the Precision@3 or Precision@5 will decrease—even if the overall result set is strong.

For instance, if all three relevant chunks are in positions 6, 7, and 8, then Precision@3 becomes 0. This ranking quality is especially critical in healthcare systems, where users may rely on just the first few pieces of information they see—particularly in urgent situations.

**Contextual Recall**

Contextual recalllooks at the flip side of precision. Instead of asking how many retrieved chunks were relevant, it asks:  
**“Of all the documents that could have helped answer the question correctly, how many did the retriever actually return?”**

This metric helps us understand how complete the system’s output is. It answers the key question:  
**“Did the system miss anything important?”**

Contextual recall is computed using the formula:

This formula tells us, out of all the truly relevant chunks that exist in the entire dataset, how many did the system manage to retrieve.

Query

Example

Result

Why this matters

What are the known limitations of continuous glucose monitors (CGMs) in diabetic care?

**Recall@K**

Now, let’s look at**recall with a cutoff,**commonly referred to as Recall@K. This metric measures how many of the known relevant chunks the system was able to retrieve **within the top K results**.

It answers the question:  
**“Out of all the important chunks that exist, how many did the system bring to the top of the list?”**

This matters because, in real-world applications such as clinical diagnostics, research synthesis, or regulatory review, the **completeness at the top of the list** can impact safety, compliance, and decision-making quality.

Recall@K is computed using the formula:

This formula tells us, out of all the truly relevant chunks that exist in the entire dataset, how many did the system manage to retrieve within the top K results.

Query

Example

Result

Why this matters

What are the known limitations of continuous glucose monitors (CGMs) in diabetic care?

**Contextual Relevancy**

Contextual relevancy measures how well a retrieved chunk matches the **meaning and intent** of a user’s  
query—even if it doesn’t use the exact same words. This is about understanding the semantics of the text, not just the surface text. A chunk is contextually relevant if it helps answer the question or adds meaningful information that supports the user's need.

This becomes especially important in healthcare, where **multiple terms can describe the same concept**. For example, “high blood sugar” and “hyperglycemia” mean the same thing. A system that only looks for keyword matches might miss relevant content just because it uses different terms. A system that understands **contextual relevancy,**however, would connect both terms to the same concept.

Let’s look at an example to understand this in detail:

Query

Example

Result

Why this matters

How do wearable devices help detect abnormal heart rhythms in elderly patients?

To evaluate contextual relevancy, we often use:

* **Embedding similarity** (e.g. cosine similarity using sentence embeddings)
* **LLM-based scoring,** where the model is asked: “Does this chunk support or meaningfully answer the question?”
* **Human evaluation,** where reviewers rate how well a chunk addresses the query’s intent

This helps understand whether the retrieval system is accurate as well as truly helpful for downstream tasks like question answering or summarization.

**Summary**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Measures** | **Focus** |
| **Precision** | Accuracy of retrieved results | Correctness |
| **Recall** | Coverage of relevant results | Completeness |
| **Relevancy** | Semantic fit | Intent alignment |

Additional content has been loaded

SKIP TO LESSON

EXIT COURSE

**Lesson 2 of 3**

[**Generative AI Solutions - Evaluation of RAG Systems**](https://uhg.edcast.com/lxphubservice/content/private/300439/cninv-21856984-2db8-4b00-97b1-d56e5dbf80c0/scormcontent/index.html#/)

[**1 of 3 — Evaluating the Retriever Component**](https://uhg.edcast.com/lxphubservice/content/private/300439/cninv-21856984-2db8-4b00-97b1-d56e5dbf80c0/scormcontent/index.html#/lessons/n_beY42tZUFG58hqE_oeAp7o7HYc3hi2)

**Evaluating the Generator Component**

**Lesson 2 of 3**

When large language models (LLMs) generate answers—such as in a medical chatbot, a clinical question-answering system, or a health-focused search engine—the response must be accurate, relevant, and grounded in real information.

This is especially important in healthcare, where people rely on AI systems to support decisions that directly impact patient safety and care. Let’s take an example from medical IoT (Internet of Things) devices. Imagine a hospital technician asks an AI assistant:  
*“How can we reduce false alarms from wearable ECG sensors in a remote cardiac monitoring system?”*

If the model replies with a general explanation of how ECG devices work, but doesn’t mention anything about reducing noise, improving signal quality, using machine learning filters, or adjusting alert thresholds, the answer might sound medically accurate—but it’s not actually useful. Even worse, if the model incorrectly suggests that turning off alerts is a solution, it could lead to missed cardiac events or patient safety risks.

This example shows why we need to evaluate whether the answer is **relevant to the specific question, factually grounded,** and**free from unsupported or misleading suggestions**.

In this section, you’ll learn four important ways to evaluate LLM-generated content:

* **Answer Relevancy** – Does the answer directly respond to the question?
* **Faithfulness** – Is the answer based on correct facts or retrieved sources?
* **Hallucination Check** – Did the model include anything that wasn’t found in the source?
* **G-Eval** – Can we use another LLM to analyze and explain how good the answer is?

By learning these techniques, you’ll be able to build and evaluate AI systems that provide **accurate, safe,** and **practically useful** answers.

**Answer Relevancy – LLM-Based**

Answer relevancy refers to how well a generated response aligns with the intent and meaning of the user’s query. Beyond being factually correct, an answer must directly address what the user is asking. This is especially important in technical healthcare contexts—such as working with medical IoT devices—where precision and contextual understanding are critical.

To evaluate answer relevancy, we can use a large language model (LLM) as a judge. The model is prompted with both the query and the generated answer, and then asked:  
**“Does this answer fully, partially, or not at all respond to the question?”**  
  
A rating (e.g. on a scale from 1 to 5) can also be requested along with a brief explanation of the judgment. Let's consider an example:

1. Click to flip

Query

1. Click to flip

Generated Answer

At first glance, this answer appears correct—it discusses heart rate monitors and data transmission. However, it fails to address the actual question, which specifically asks about security best practices for handling data from these IoT devices.

**A relevant answer should mention topics such as:**

data encryption, HIPAA compliance, secure transmission protocols (e.g. TLS/SSL), device authentication, or anomaly detection to identify unauthorized access.

In this case, an LLM used for evaluation would likely rate the answer as **“not relevant”** or **“minimally relevant”**because, while it discusses the device, it completely overlooks the core focus on data security.

This distinction matters. These devices continuously collect and transmit sensitive health data, and vague or off-target answers can lead to poor system design or regulatory non-compliance. That’s why answer relevancy is critical—especially when building tools for developers, clinicians, or IT professionals who depend on precise, actionable information.

Using an LLM to assess this alignment helps ensure that systems deliver useful, query-specific responses that meet clinical, technical, and regulatory requirements.

Answer Relevancy (LLM-Based) can be measured using DeepEval package in Python. It measures how closely the answer aligns with what was asked. It looks at whether the answer:

* Directly addresses the query
* Is factually correct
* Stays on topic and is contextually appropriate

DeepEval provides a relevancy score based on the LLM’s judgment of how well the answer fits the question.

**Why should you use it?**

**Faithfulness**

Faithfulnessrefers to whether a generated answer is factually accurate based on the information retrieved by the system or drawn from trusted sources such as clinical guidelines or medical databases. In other words, a faithful answer **stays true to the facts**—it does not invent or distort information. This is especially critical in healthcare, where inaccuracies can lead to unsafe decisions.

Let’s consider an example:

1. Click to flip

Query

1. Click to flip

Retrieved Source

1. Click to flip

Generated Answer

At first glance, the answer may sound helpful—but it is **not faithful to the source**. Nowhere in the retrieved content does it claim that hospital visits can be eliminated or that these devices can autonomously adjust oxygen therapy. In reality, most wearable SpO₂ monitors are passive monitoring tools, not automated treatment devices.

This kind of error—introducing unsupported claims—can be dangerous. It may mislead patients or clinicians about the capabilities of the technology, potentially resulting in under-monitoring, delayed interventions, or over-reliance on automation.

**A faithful version of the answer would stick to what’s in the source:**

“Wearable oxygen monitors support COPD management by helping detect low oxygen levels early and reducing hospital visits when combined with professional oversight.”

This distinction matters because even small factual inaccuracies can have serious consequences. Evaluating faithfulness helps ensure that AI-generated responses do not misrepresent the facts, even if they sound plausible.

To assess faithfulness, you can prompt an LLM with a question like:  
**“Does this answer accurately reflect the retrieved source? Highlight anything that is not supported.”**

DeepEval can be used to measure faithfulness. It checks whether the answer sticks to the facts in the source context and avoids adding any unsupported or misleading information. DeepEval evaluates each answer and gives a faithfulness score based on how closely it stays aligned with the context.

**Why should you use it?**

**Hallucination Check**

Hallucination occurs when a language model generates content that is not based on retrieved sources or verified medical knowledge. This could include fabricated facts, incorrect assumptions, or even fake references. In healthcare applications—especially those involving patient data, clinical tools, or medical devices—hallucinations can be misleading or even harmful.

Let’s consider an example:

1. Click to flip

Query

1. Click to flip

Retrieved Source

1. Click to flip

Generated Answer

At first glance, the answer may sound innovative—but this is a hallucination. The retrieved source only states that smart inhalers track usage and support treatment adherence. It does not mention automatic drug administration or real-time attack detection. In reality, most smart inhalers currently available are monitoring tools, not autonomous treatment systems. The model has invented a capability that the devices do not possess.

This type of hallucination can mislead users—whether patients, clinicians, or healthcare developers—into believing the technology is more advanced or autonomous than it is. In real-world settings, this could result in dangerous over-reliance on the device or inadequate emergency preparedness.

This matters because hallucinations can create false expectations and introduce clinical risks. In regulated domains like healthcare, fabricated claims—especially about medical devices—can lead to non-compliance, loss of trust, or even patient harm. Users may assume that if the AI said it, it must be true—especially when the output sounds confident.

**To detect hallucinations, you can prompt an LLM (or a human reviewer) with:**

“Which parts of this answer are not supported by the retrieved documents?  
Use binary labeling:Supported / Unsupported. Highlight unsupported parts for review or retraining."

DeepEval can be used to detect hallucinations. It evaluates whether the answer is fully grounded in the context or if the model is generating facts or details that don’t exist in the source material. Hallucination occurs when the model includes information that is:

* Not present in the context
* Factually incorrect
* Misleading or invented

DeepEval flags such responses and provides a hallucination score, which helps in tracking how often the model is “making things up”.

**Why should you use it?**

**G-Eval – Custom LLM as Judge (with Chain-of-Thought Reasoning)**

**G-Eval** stands for **Generative Evaluation**. It refers to using a customized large language model (LLM) to act as a judge that scores and explains the quality of generated answers. What makes G-Eval special is that it uses **chain-of-thought (CoT) reasoning**—in other words, the model doesn’t just give a score; it walks through its reasoning step by step, just like a human reviewer would.

Instead of evaluating only one parameter, G-Eval can assess answers across **multiple dimensions,** such as:

* Relevancy – Does the answer match the question?
* Coherence – Is the answer logically consistent and well-organized?
* Factual Accuracy – Is it correct based on source data?
* Tone – Is it professional, empathetic, or appropriate for the audience?

Let’s consider an example:

1. Click to flip

Query

1. Click to flip

Generated Answer A

1. Click to flip

Generated Answer B

Both answers sound okay at first glance. But which one is better? That’s where **G-Eval**comes in. By prompting the LLM with a structured evaluation task, we can ask it to judge both responses on the key criteria above—but also to explain why.

**G-Eval Prompt Example:**

“Given the query and two model responses, evaluate both answers in terms of: (1) Relevancy, (2) Factual Accuracy, and (3) Coherence. For each, explain your reasoning and assign a score from 1 to 5.”

The model might reason like this:

* Answer A is directly relevant to the query and explains how ECG devices help detect AFib. It stays factual and clearly connects detection to timely action. Score: 5/5.
* Answer B includes some accurate information but focuses more on device connectivity and general AFib symptoms. It doesn’t explain how the device helps in detecting AFib. Score: 3/5.

G-Eval matters because it gives a **structured, interpretable way** to evaluate outputs using the model itself. It simulates expert review but at scale, making it useful during system development, A/B testing, or QA. It also encourages consistency in evaluation and helps uncover why one answer is better than another.

**How to use G-Eval with CoT reasoning?**

1. Use a prompt that asks the LLM to score and explain its reasoning.
2. Include the query, retrieved sources (if applicable), and generated answer(s) in the input.
3. Ask for both scores and justifications so you can learn from the model’s thinking process.

This approach works especially well when you're evaluating responses for medical support tools, IoT integration systems, or patient education platforms, where correctness and clarity are both critical.

G-Eval uses a large language model to assess the entire response based on multiple quality dimensions — like relevance, fluency, coherence, factual accuracy, and completeness. G-Eval doesn’t focus on one metric — it gives a holistic judgment of how good the answer is.

Depending on the setup, G-Eval can evaluate:

1. Click to flip

How well the answer addresses the query

1. Click to flip

If the language is clear and grammatically correct

1. Click to flip

Whether the response is logically structured

1. Click to flip

If the facts are correct and supported

1. Click to flip

How complete or thorough the answer is

**Why should you use it?**

**Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **What It Measures** | **Evaluation Approach** | **Risk if Ignored** |
| **Answer Relevancy** | How well the answer matches the user's question | Use an LLM to judge if the answer fully, partially, or doesn’t address the query | May return off-topic or unhelpful answers that sound fluent but don’t meet the user’s need |
| **Faithfulness** | Whether the answer stays true to source data | Compare answer to retrieved content or verify using an LLM | Can lead to incorrect or misleading claims, even if the answer sounds confident |
| **Hallucination** | Whether the model made up unsupported content | Use LLM or rule-based checks to flag unsupported statements | May generate fake facts, harmful advice, or unrealistic system capabilities |
| **G-Eval (CoT)** | Overall answer quality across multiple dimensions | Use an LLM with step-by-step reasoning to score relevance, accuracy, tone, etc. | Can miss deeper quality issues like subtle inaccuracies or poor reasoning in responses |

[**3 of 3 — Evaluation Metrics**](https://uhg.edcast.com/lxphubservice/content/private/300439/cninv-21856984-2db8-4b00-97b1-d56e5dbf80c0/scormcontent/index.html#/lessons/Ha7Z-FRC0nOp6bMFU3-VRMushosniCYK)

test\_case = LLMTestCase(

input=response['question'],

actual\_output=response['AI\_generated\_response'],

expected\_output=human\_answer,

retrieval\_context=retrieved\_context

)

metric = ContextualRelevancyMetric(

threshold=0.6,

model=wrapped\_model,

include\_reason=True,

verbose\_mode=True

)

result = evaluate([test\_case], [metric])