

Date
22 Aug 2023

Authors

Kim Martineau

Topics

[AI](#)
[Explainable AI](#)
[Generative AI](#)
[Natural Language Processing](#)
[Trustworthy Generation](#)

Share



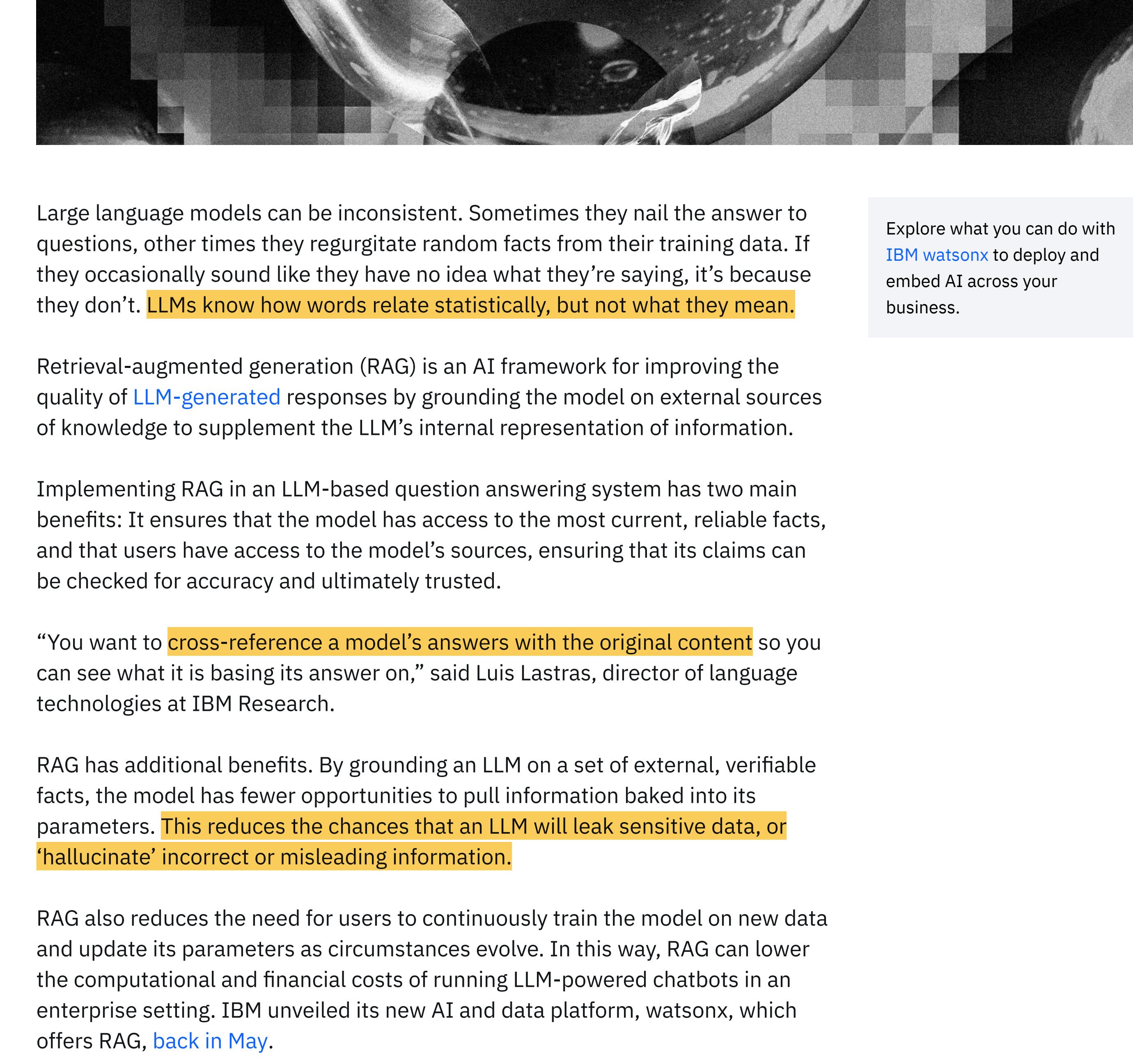
Explainer

5 minute read

Tldr; ground-truthing LLMs

What is retrieval-augmented generation?

RAG is an AI framework for retrieving facts from an external knowledge base to ground large language models (LLMs) on the most accurate, up-to-date information and to give users insight into LLMs' generative process.



Large language models can be inconsistent. Sometimes they nail the answer to questions, other times they regurgitate random facts from their training data. If they occasionally sound like they have no idea what they're saying, it's because they don't. [LLMs know how words relate statistically, but not what they mean.](#)

Explore what you can do with [IBM watsonx](#) to deploy and embed AI across your business.

Retrieval-augmented generation (RAG) is an AI framework for improving the quality of [LLM-generated](#) responses by grounding the model on external sources of knowledge to supplement the LLM's internal representation of information.

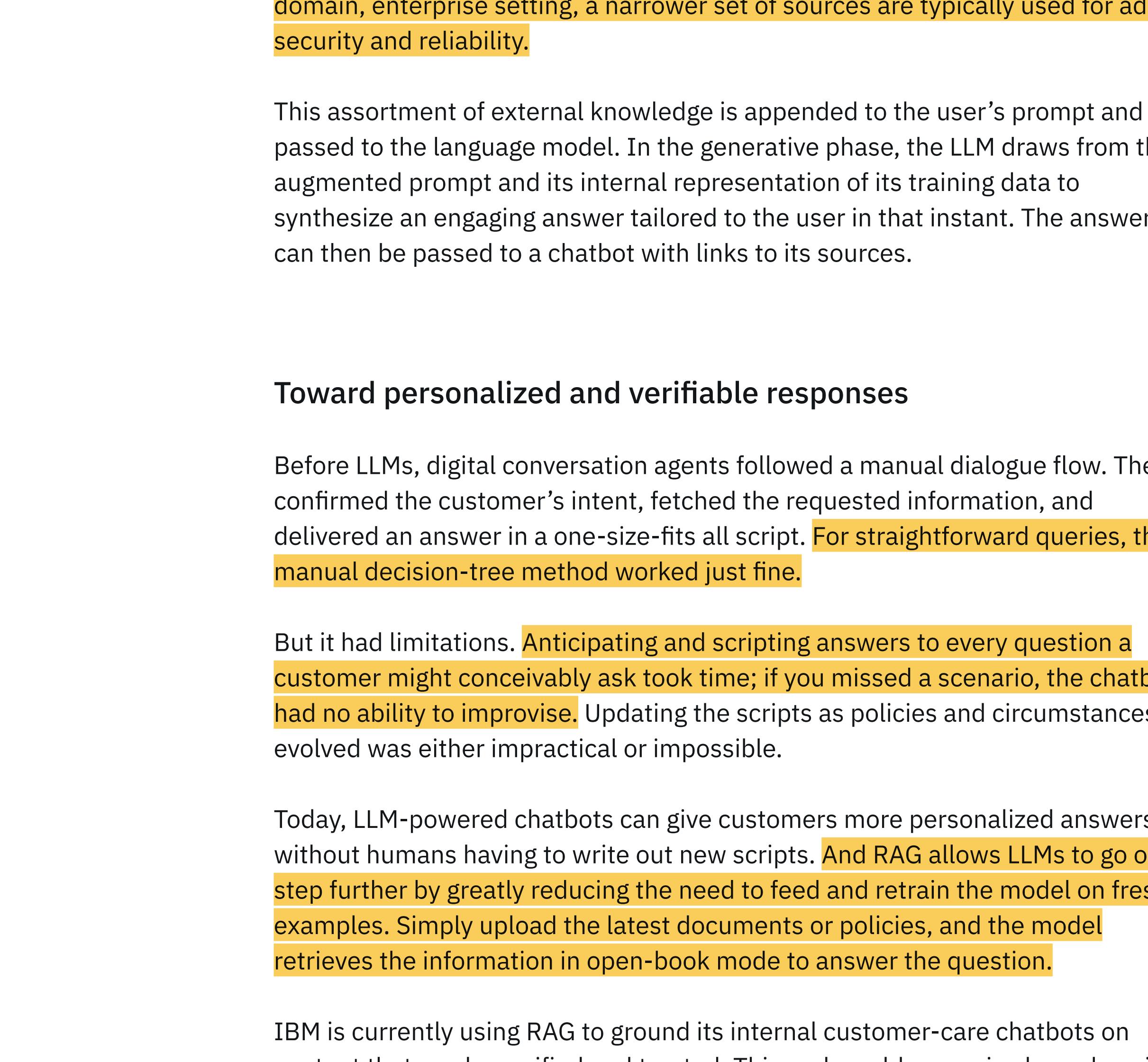
Implementing RAG in an LLM-based question answering system has two main benefits: It ensures that the model has access to the most current, reliable facts, and that users have access to the model's sources, ensuring that its claims can be checked for accuracy and ultimately trusted.

"You want to [cross-reference a model's answers with the original content](#) so you can see what it is basing its answer on," said Luis Lastras, director of language technologies at IBM Research.

RAG has additional benefits. By grounding an LLM on a set of external, verifiable facts, the model has fewer opportunities to pull information baked into its parameters. [This reduces the chances that an LLM will leak sensitive data, or 'hallucinate' incorrect or misleading information.](#)

RAG also reduces the need for users to continuously train the model on new data and update its parameters as circumstances evolve. In this way, RAG can lower the computational and financial costs of running LLM-powered chatbots in an enterprise setting. IBM unveiled its new AI and data platform, [watsonx](#), which offers RAG, [back in May](#).

An overview from IBM expert Marina Danilevsky



Hear from other AI experts

- Dmitri Krotov is on a quest to [improve AI](#).
- Ruchir Puri leads the charge on [code modernization](#).
- Kush Varshney explains how [governance](#) is applied throughout the AI pipeline.

An 'open book' approach to answering tough questions

[Underpinning all foundation models](#), including LLMs, is an AI architecture known as the transformer. It turns heaps of [raw data](#) into a compressed representation of its basic structure. Starting from this raw representation, a foundation model can be adapted to a variety of tasks with some additional fine-tuning on labeled, domain-specific knowledge.

But fine-tuning alone rarely gives the model the full breadth of knowledge it needs to answer highly specific questions in an ever-changing context. In a [2020 paper](#), Meta (then known as Facebook) came up with a framework called [retrieval-augmented generation](#) to [give LLMs access to information beyond their training data](#). RAG allows LLMs to build on a specialized body of knowledge to answer questions in more accurate way.

"It's the difference between an open-book and a closed-book exam," Lastras said. "In a RAG system, you are asking the model to respond to a question by browsing through the content in a book, as opposed to trying to remember facts from memory."

As the name suggests, RAG has two phases: retrieval and content generation. In the retrieval phase, algorithms search for and retrieve snippets of information relevant to the user's prompt or question. In an open-domain, consumer setting, those facts can come from indexed documents on the internet; in a closed-domain, enterprise setting, a narrower set of sources are typically used for added security and reliability.

This assortment of external knowledge is appended to the user's prompt and passed to the language model. In the generative phase, the LLM draws from the augmented prompt and its internal representation of its training data to synthesize an engaging answer tailored to the user in that instant. The answer can then be passed to a chatbot with links to its sources.

Toward personalized and verifiable responses

Before LLMs, digital conversation agents followed a manual dialogue flow. They confirmed the customer's intent, fetched the requested information, and delivered an answer in a one-size-fits-all script. [For straightforward queries, this manual decision-tree method worked just fine.](#)

But it had limitations. [Anticipating and scripting answers to every question a customer might conceivably ask took time](#); if you missed a scenario, the chatbot had no ability to improvise. Updating the scripts as policies and circumstances evolved was either impractical or impossible.

Today, LLM-powered chatbots can give customers more personalized answers without humans having to write out new scripts. [And RAG allows LLMs to go one step further by greatly reducing the need to feed and retrain the model on fresh examples](#). Simply upload the latest documents or policies, and the model retrieves the information in open-book mode to answer the question.

IBM is currently using RAG to ground its internal customer-care chatbots on content that can be verified and trusted. This real-world scenario shows how it works: An employee, Alice, has learned that her son's school will have early dismissal on Wednesdays for the rest of the year. She wants to know if she can take vacation in half-day increments and if she has enough vacation to finish the year.

To craft its response, the LLM first pulls data from Alice's HR files to find out how much vacation she gets as a longtime employee, and how many days she has left for the year. It also searches the company's policies to verify that her vacation can be taken in half-days. These facts are injected into Alice's initial query and passed to the LLM, which generates a concise, personalized answer. A chatbot delivers the response, with links to its sources.

Teaching the model to recognize when it doesn't know

Customer queries aren't always this straightforward. They can be ambiguously worded, complex, or require knowledge the model either doesn't have or can't easily parse. These are the conditions in which LLMs are prone to making things up.

"Think of the model as an overeager junior employee that blurts out an answer before checking the facts," said Lastras. "Experience teaches us to stop and say when we don't know something. But LLMs need to be explicitly trained to recognize questions they can't answer."

In a more challenging scenario taken from real life, Alice wants to know how many days of maternity leave she gets. A chatbot that does not use RAG responds cheerfully (and incorrectly): "Take as long as you want."

Maternity-leave policies are complex, in part, because they vary by the state or country of the employee's home-office. When the LLM failed to find a precise answer, it should have responded, "I'm sorry, I don't know," said Lastras, or asked additional questions until it could land on a question it could definitively answer. Instead, it pulled a phrase from a training set stocked with empathetic, customer-pleasing language.

With enough fine-tuning, an LLM can be trained to pause and say when it's stuck. But it may need to see thousands of examples of questions that can and can't be answered. Only then can the model learn to identify an unanswerable question, and probe for more detail until it hits on a question that it has the information to answer.

RAG is currently the best-known tool for grounding LLMs on the latest, verifiable information, and lowering the costs of having to constantly retrain and update them. RAG depends on the ability to enrich prompts with relevant information contained in vectors, which are mathematical representations of data. [Vector databases](#) can efficiently index, store and retrieve information for things like recommendation engines and chatbots. But RAG is imperfect, and many interesting challenges remain in getting RAG done right.

At IBM Research, we are focused on innovating at both ends of the process: retrieval, how to find and fetch the most relevant information possible to feed the LLM; and generation, how to best structure that information to get the richest responses from the LLM.

See how we're applying AI research

- We are setting out to reduce the environmental [impact of PFAS](#).
- We are developing methods to [improve chatbots](#).
- We are co-creating an AI foundation model for [weather and climate prediction](#).

Subscribe to our Future Forward

newsletter and stay up to date

on the latest research news

Subscribe to our newsletter

News

IBM Granite leads open-source LLMs on API calling

Kim Martineau

08 Aug 2024

[AI](#) [Conversational AI](#) [Foundation Models](#)

News

IBM and RPI researchers demystify in-context learning in large language...

Peter Hess

25 Jul 2024

[AI](#) [AI Transparency](#) [Explainable AI](#)

News

What's an LLM context window and why is it getting larger?

Kim Martineau

24 Jul 2024

[AI](#) [AI for Code](#) [Generative AI](#) [Granite](#)

News

Meet AI-Hilbert, a new algorithm for transforming scientific discovery

Peter Hess

19 Jul 2024

[AI](#) [Exploratory Science](#) [Physical Sciences](#)

IBM

Focus areas

Semiconductors

Artificial Intelligence

Quantum Computing

Hybrid Cloud

Quick links

About

Publications

Blog

Events

Directories

Topics

People

Projects

Follow us

Newsletter

X

LinkedIn

YouTube

Contact IBM

Privacy

Terms of use

Accessibility

Cookie Preferences

News

IBM Granite leads open-source LLMs on API calling

Kim Martineau

08 Aug 2024

[AI](#) [Conversational AI](#) [Foundation Models](#)

News

IBM and RPI researchers demystify in-context learning in large language...

Peter Hess

25 Jul 2024

[AI](#) [AI Transparency](#) [Explainable AI](#)

News

What's an LLM context window and why is it getting larger?

Kim Martineau

24 Jul 2024

[AI](#) [AI for Code](#) [Generative AI](#) [Granite](#)

News

Meet AI-Hilbert, a new algorithm for transforming scientific discovery

Peter Hess

19 Jul 2024

[AI](#) [Exploratory Science](#) [Physical Sciences](#)