

Question-driven content design (QDCD)

Sarah Packowski

AI ContentOps Architect, IBM



Agenda

1. How content fuels RAG and AI agents
2. Traditional content design process
3. How the traditional process fails RAG, AI agents
4. Solution: question-driven content design
5. Hands-on demonstration

1/5 **How content fuels RAG and AI agents**

RAG and agentic solutions cannot work without pulling in accurate, up-to-date, domain-specific content that writers create.

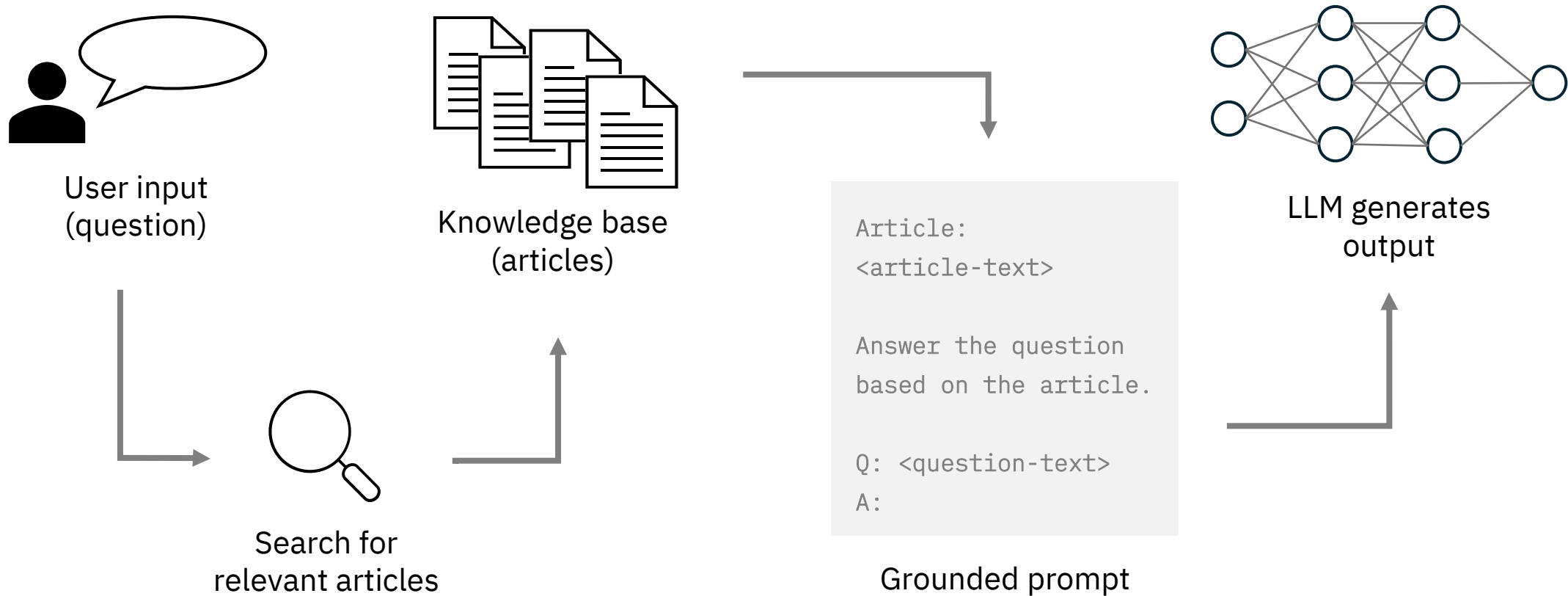
1/5 How content fuels RAG and AI agents

RAG and agentic solutions cannot work without pulling in accurate, up-to-date, domain-specific content that writers create.

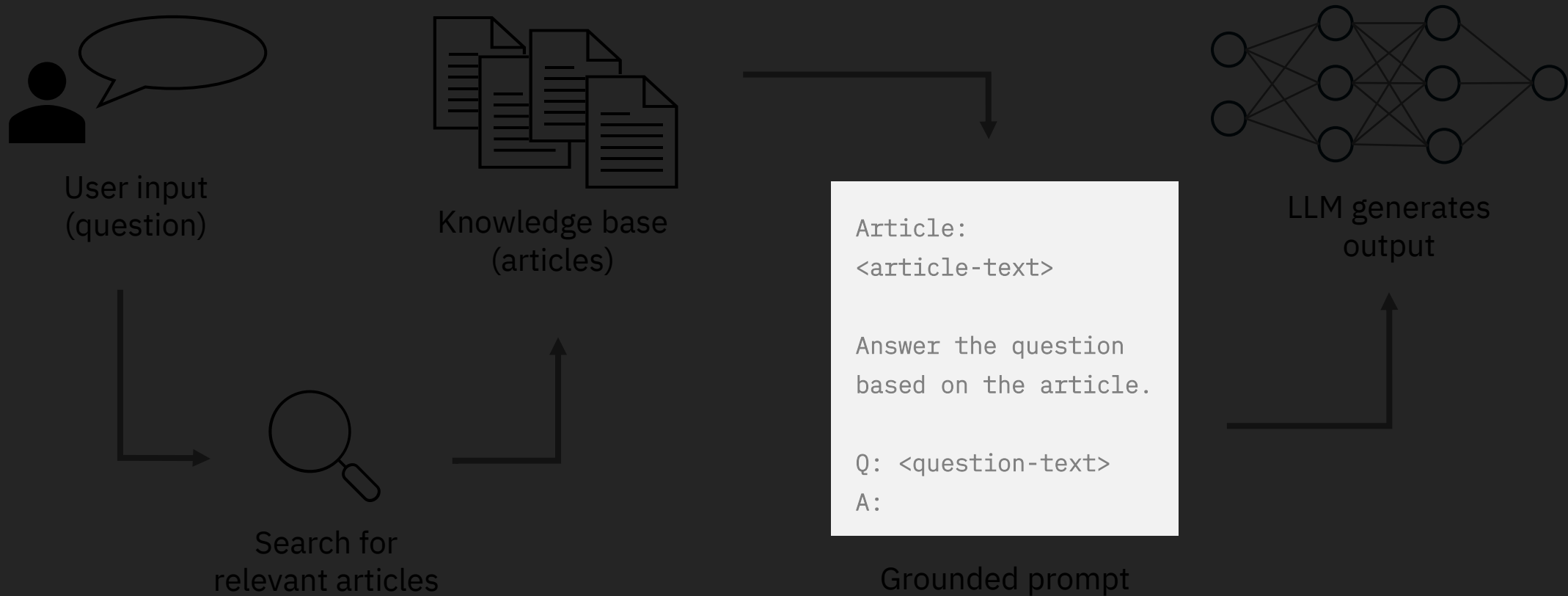


Thing 1 to remember

Retrieval-augmented generation (RAG)



Retrieval-augmented generation (RAG)



Scenario



Image generated using ChatGPT

You have invented a new hand-held writing implement, called the *carbonWrite 9000*



IBM watsonx

[Projects](#) / [Experiments with prompts](#) / Prompt Lab

Model: flan-t5-xxl-11b ▼

Question: What kind of thing is the carbonWrite 9000?

Answer: printer

Stop reason: End of sequence token encountered
Tokens: 18 input + 2 generated = 20 out of 4096
Time: 0.4 seconds

[Clear output](#)

Generate

watsonx.ai



Model: flan-t5-xxl-11b ▾

Article:

Congratulations on purchasing the carbonWrite 9000 pencil!

Once you have sharpened the end, you can use the pencil to write and draw on a variety of surfaces, including paper and cardboard.

Answer the following question using only information from the article.

Question: What kind of thing is the carbonWrite 9000?

Answer: pencil

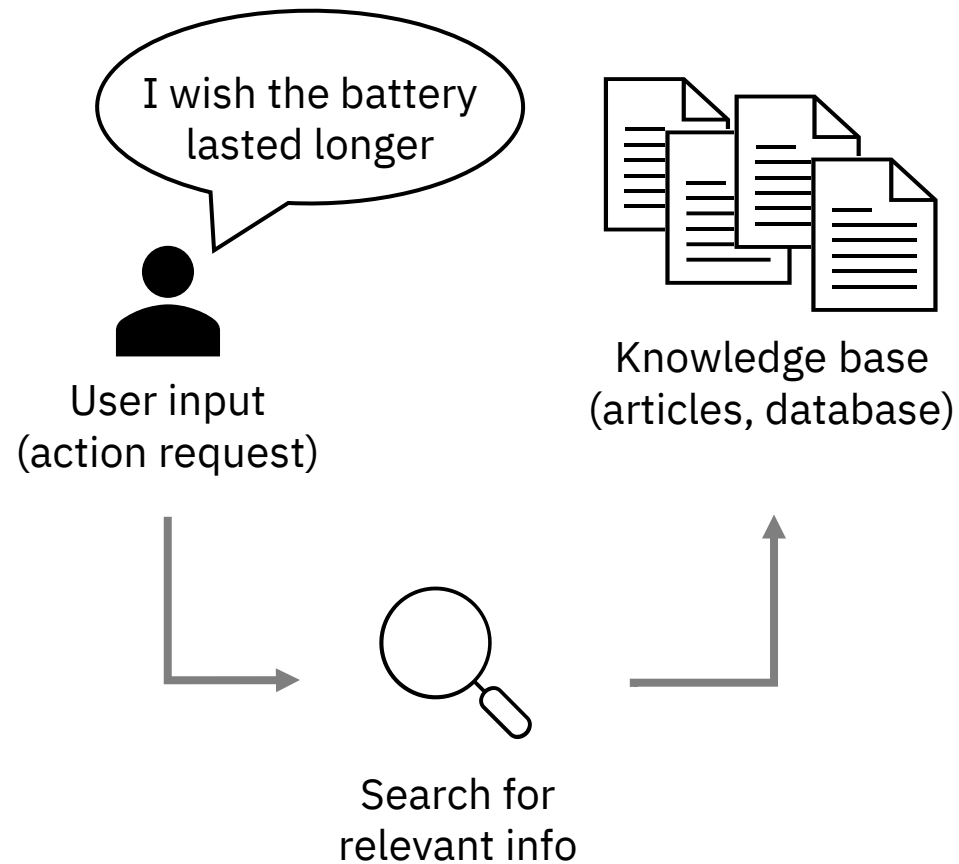
Stop reason: End of sequence token encountered

Tokens: 18 input + 2 generated = 20 out of 4096

Time: 0.4 seconds

[Clear output](#) [Generate](#)

AI agents



What is the command to achieve the goal in the following user input?

API reference:

To configure the carbonWrite 9000 battery mode, call battery_config.

Syntax

battery_config [performance | longevity]

Example 1: For performance

battery_config performance

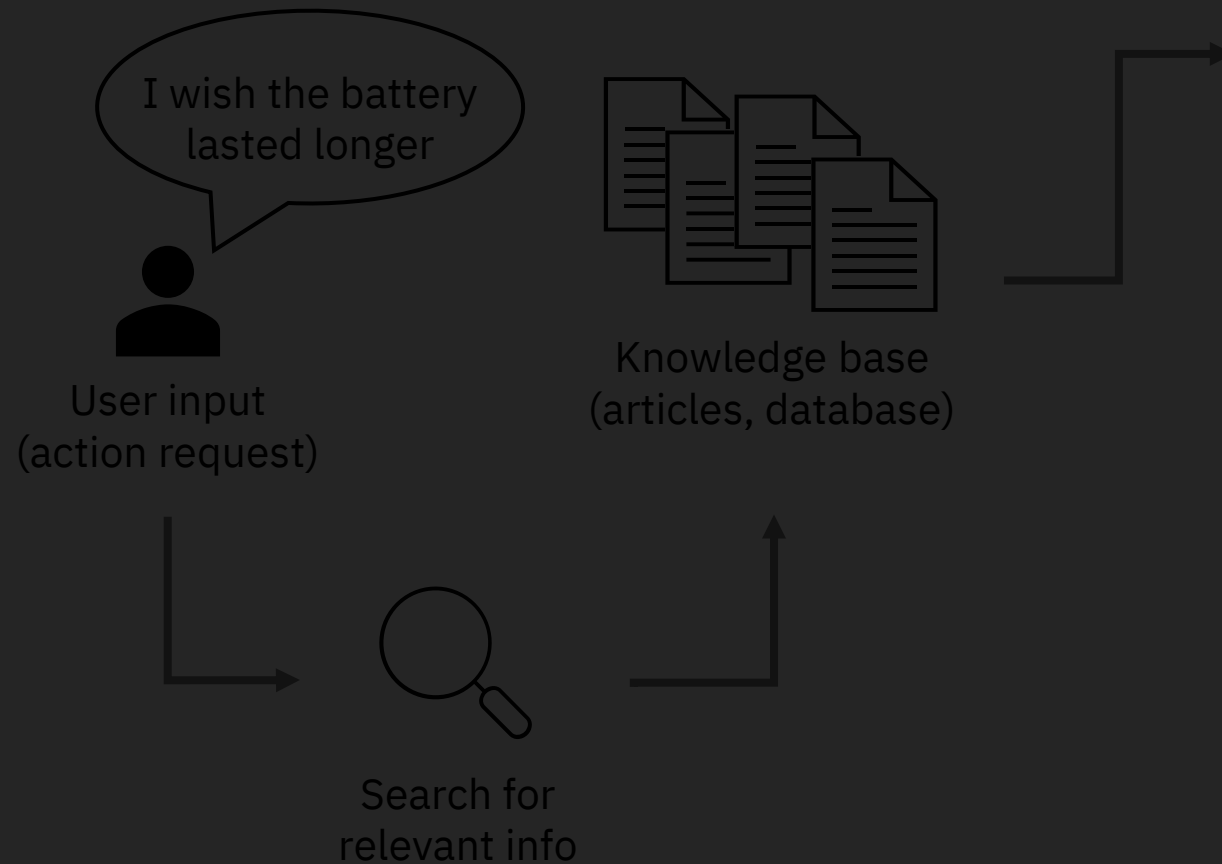
Example 2: For long battery life

battery_config longevity

User input: I wish the battery lasted longer

A: **battery_config longevity**

AI agents



What is the command to achieve the goal in the following user input?

API reference:

To configure the carbonWrite 9000 battery mode, call battery_config.

Syntax

```
battery_config [ performance | longevity ]
```

Example 1: For performance

```
battery_config performance
```

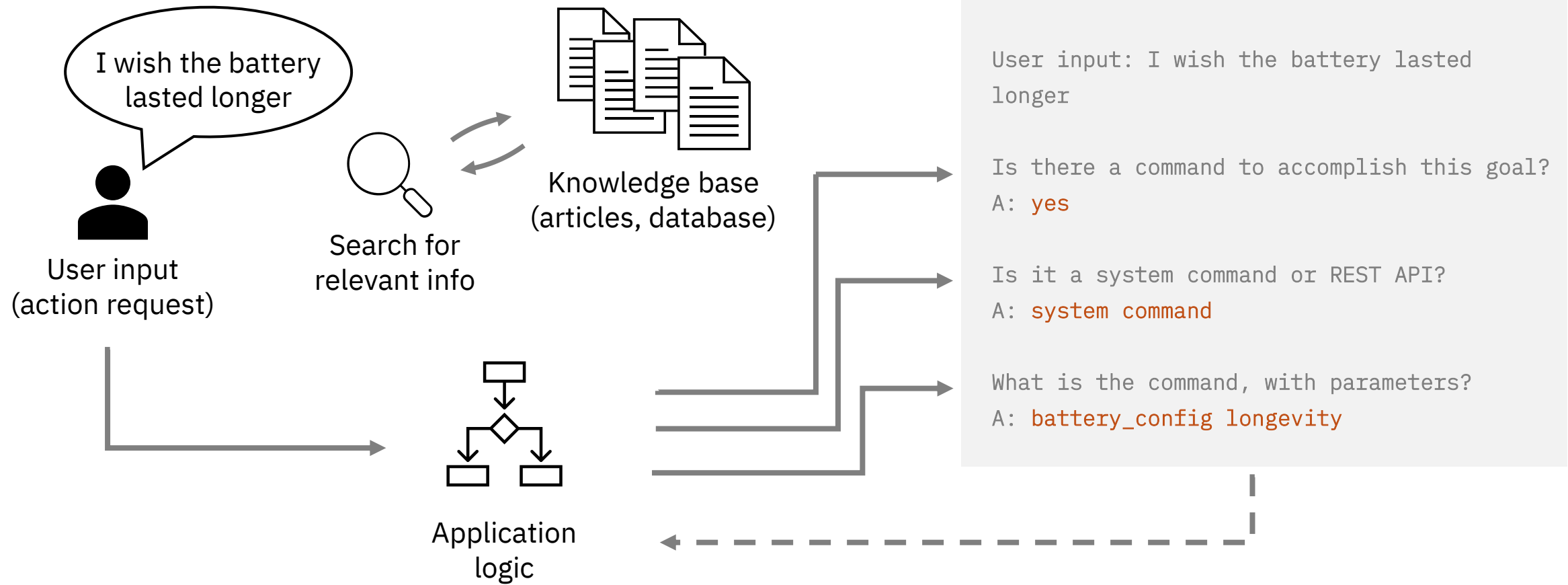
Example 2: For long battery life

```
battery_config longevity
```

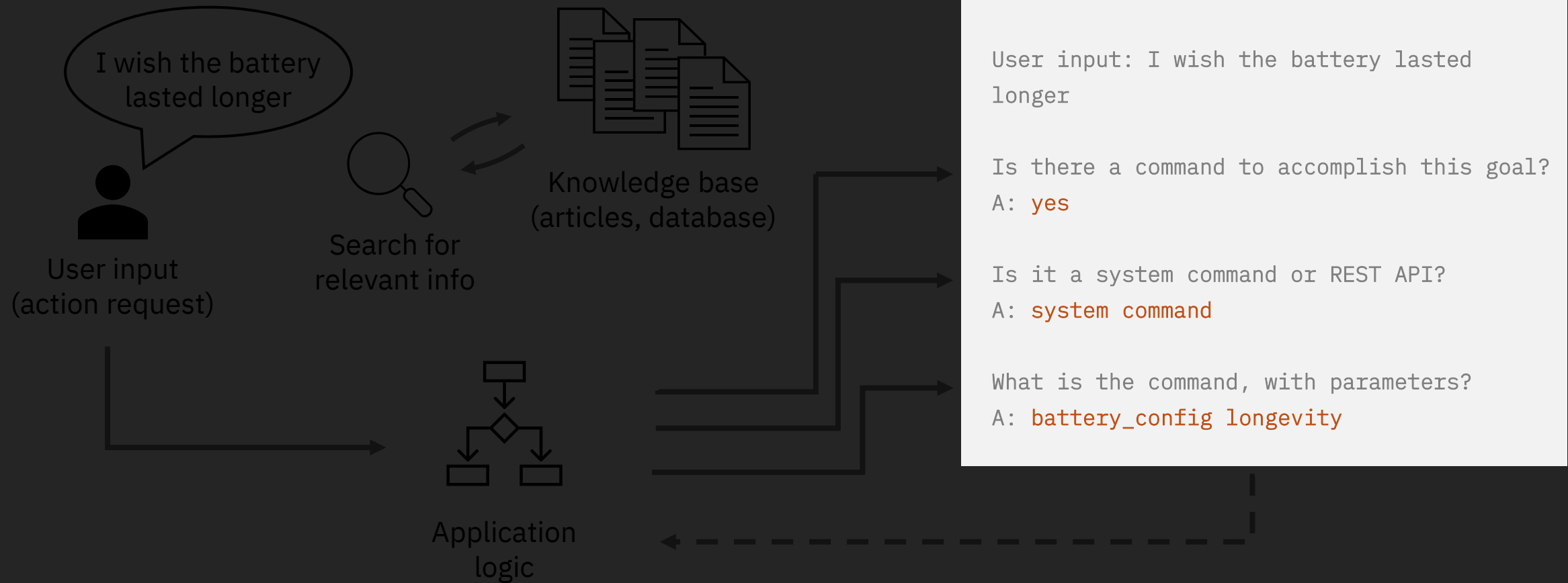
User input: I wish the battery lasted longer


A: `battery_config longevity`

AI agents



AI agents



 IBM watsonx

Projects / Experiments with prompts / Prompt Lab

Model: flan-t5-xxl-11b

The goal is: I wish the battery on this thing lasted longer

Information related to this goal is in the following article:

You can configure the carbonWrite 9000 battery in one of two modes:

- High performance, for faster response times and brighter light
- Long life, to extend the battery life as long as possible

To configure the battery mode, call the battery_config administrative function from a system command line.

Command syntax:

```
battery_config [mode]
```

Parameters:

[mode] is either "performance" or "longevity"

Example 1: Configuring for performance

```
battery_config performance
```

Example 2: Configuring for battery life

```
battery_config longevity
```

Is there a command or REST API that can be used to accomplish this goal, yes or no?

yes


Is it a system command or REST API?

system command

What is the command to call, with parameters?

```
battery_config longevity
```

Stop reason: End of sequence token encountered
Tokens: 205 input + 5 generated = 210 out of 4096
Time: 0.4 seconds

Clear output 

Generate →

1/5 How content fuels RAG and AI agents

RAG and agentic solutions cannot work without pulling in accurate, up-to-date, domain-specific content that writers create.



Thing 1 to remember

2/5 Traditional content design process

1. Identify *concepts* users need to understand, *tasks* users will perform, and *reference* information users need to perform those tasks
2. Break the information into pieces, so you can write an individual article for each piece
3. Map out how all the information pieces will fit together

3/5 **How the traditional process fails RAG, AI agents**

- Real user questions contain “incorrect” terms
- Real user questions are based on “misunderstanding”
- Real users ask about “unsupported” tasks

Reality: Unexpected questions

Pencils

Congratulations on purchasing the Carbon9000 pencil!

Once you have sharpened the end, you can use the pencil to write and draw on a variety of surfaces.

If you purchased the optional rubOut eraser feature, you can erase what you've written with the pencil.

Does the pen write on cardboard?

Can you stir your coffee with it?

Will the ink leak in my pocket?

Can you rewind a cassette tape with it?

3/5 How the traditional process fails RAG, AI agents

- Real user questions contain “incorrect” terms
- Real user questions are based on “misunderstanding”
- Real users ask about “unsupported” tasks
- New AI interfaces lack context

Documentation: provides context

Set expectations,
shape user behavior:

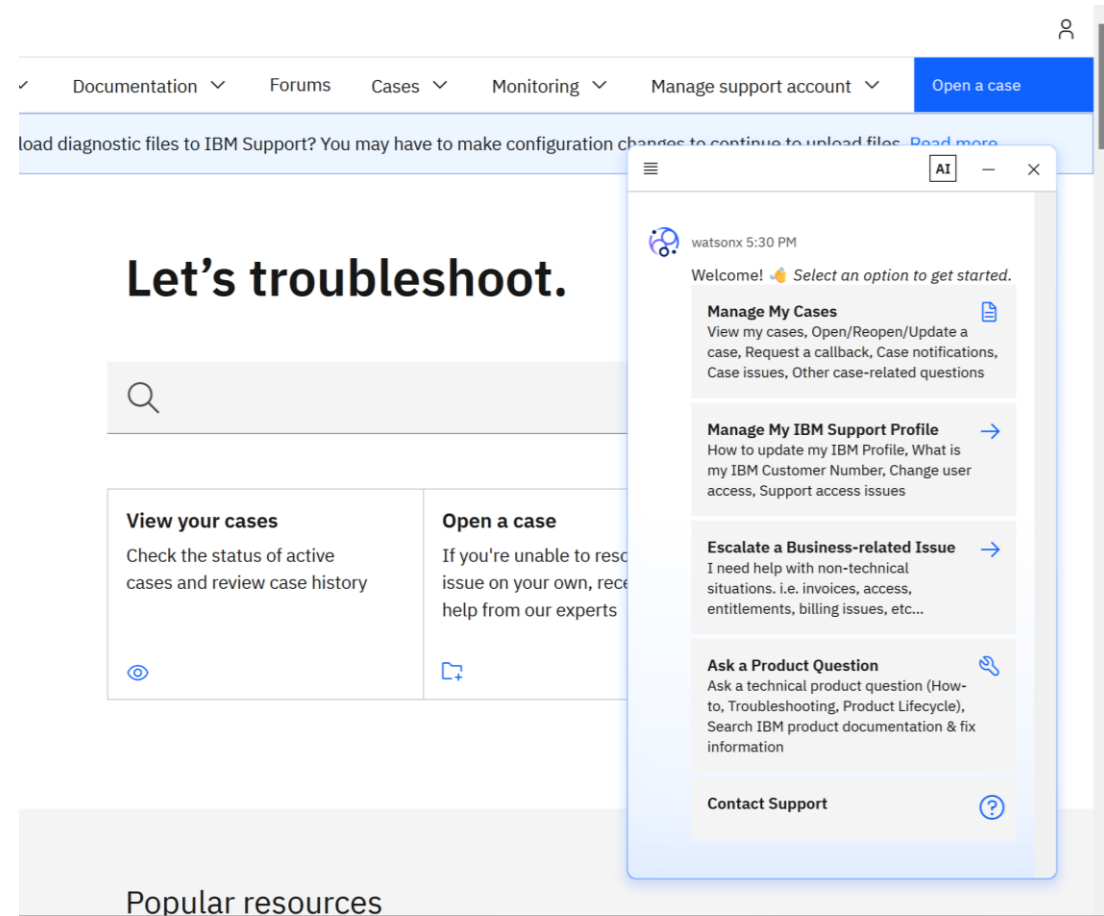
- Table of content
- Headings
- Links to details
- Diagrams

The screenshot shows the IBM watsonx documentation interface. The left sidebar contains a navigation menu with sections like Overview, Planning a generative AI solution, Getting started and tutorials, Gen AI solutions (expanded), Terms of use, Tokens, Foundation model deployment methods (expanded), Supported models, Building prompts, Tuning models, Agent Lab (beta), Coding generative AI solutions, Retrieval-augmented generation, Automating a RAG pattern with AutoAI, Preparing data, Data science solutions, Deploying AI, and Governing AI. The main content area is titled 'Developing generative AI solutions with foundation models' and includes a 'Last updated' date of Mar 14, 2025. The text explains that users can develop generative AI solutions with foundation models in IBM watsonx.ai, creating prompts to generate, classify, summarize, or extract content. It also defines foundation models as large AI models trained on terabytes of data, capable of various tasks like text, code, or image generation. A section titled 'Foundation model architecture' introduces a diagram comparing traditional AI models (Training → model → Tasks) with foundation models (Massive external data → Tasks).

Chat: providing context is difficult

Attempts to provide context:

- Introductory message
- “Tips”
- Suggested topics



Voice interface: No context

- “What can I do here?”
- “What can I ask about?”
- “What can you do?”

3/5 **How the traditional process fails RAG, AI agents**

- Real user questions contain “incorrect” terms
- Real user questions are based on “misunderstanding”
- Real users ask about “unsupported” tasks
- New AI interfaces lack context

4/5 **Solution: question-driven content design**

1. Collect real user questions
2. Test that content can answer those questions

4/5 **Solution: question-driven content design**

1. Collect real user questions
2. Test that content can answer those questions

QDCD step 1: Collect real user questions

You cannot guess

- Your guesses will be biased due to familiarity with the source material
- Your terminology, phrasing will be influenced by having worked with the content

Where to find user questions

- Forums
- User communities
- User feedback
- User research
- Internet searchers
- ...

4/5 **Solution: question-driven content design**

1. Collect real user questions
2. Test that content can answer those questions

QDCD step 2: Test content



Forward

Configure the RAG/agent solution to use the content as a knowledge base, submit real user questions, then evaluate generated answers.

Advantages

- Results exactly match run-time results

Challenges

- Setting this up is not trivial
- Solutions are not all built the same (Do you test them all?)
- Manual evaluation is time-consuming, automated evaluation can be misleading

QDCD step 2: Test content



Generate questions answered by the content, then compare the generated questions to the real user questions.

Advantages

- Can be automated
- Scales easily

Challenges

- Is an indirect measure

QDCD step 2: Test content (option 2)

<https://youtu.be/OWoHLQt0ANo>


Results

Score is the similarity (0-100) between the user question and the best-matching LLM-generated question (bold)

User question	Closest generated questions	Score ^	Match?
Do dogs and foxes get along?	<ul style="list-style-type: none">• Did the fox jump over the dog?• Was the fox fast or slow?• What did the fox do?	77	X
What color are dogs?	<ul style="list-style-type: none">• What color was the dog?• Was the dog fast or slow?• What color was the fox?	91	✓
What color are foxes?	<ul style="list-style-type: none">• What color was the fox?• What color was the dog?• Was the fox fast or slow?	94	✓
Are dogs lazy?	<ul style="list-style-type: none">• Was the dog lazy?	94	✓

Similarity threshold

90



5/5 Hands-on demonstration

bit.ly/Q-D-C-D

Lesson

Be less attached to our traditional processes and guidelines. Create content to meet readers where they are.

Lesson

Be less attached to our traditional processes and guidelines. Create content to meet readers where they are.

A yellow starburst graphic with a red outline, containing the text "Thing 2 to remember".

Thing 2 to
remember

Lesson

Desire path (desire line)

https://en.wikipedia.org/wiki/Desire_path



https://commons.wikimedia.org/wiki/File:Desire_line_-_geograph.org.uk_-_6020249.jpg



<https://www.theguardian.com/cities/2018/oct/05/desire-paths-the-illicit-trails-that-defy-the-urban-planners>

Question-driven content design

Fact : Rag, agentic solutions depend on content

Question-driven content design

Fact : Rag, agentic solutions depend on content



Question-driven content design

Fact : Rag, agentic solutions depend on content



Problem : Current content design process fails RAG, agentic solutions

Question-driven content design

Fact : Rag, agentic solutions depend on content



Problem : Current content design process fails RAG, agentic solutions

One solution : Question-driven content design

1. Collect real user questions
2. Test that content can answer those questions

Question-driven content design

Fact : Rag, agentic solutions depend on content



Problem : Current content design process fails RAG, agentic solutions

One solution : Question-driven content design

1. Collect real user questions
2. Test that content can answer those questions

New reality : We need to write for users where they are

Question-driven content design

Fact : Rag, agentic solutions depend on content



Problem : Current content design process fails RAG, agentic solutions

One solution : Question-driven content design

1. Collect real user questions
2. Test that content can answer those questions

New reality : We need to write for users where they are



Resources

bit.ly/Q-D-C-D