

Open in app ↗

Medium

 Search Write

Developing RAG Systems with DeepSeek R1 & Ollama (Complete Code Included)

Sebastian Petrus · [Follow](#)

4 min read · Jan 24, 2025



1K



22



Ever wished you could directly ask questions to a PDF or technical manual? This guide will show you how to build a **Retrieval-Augmented Generation (RAG)** system using **DeepSeek R1**, an open-source reasoning tool, and **Ollama**, a lightweight framework for running local AI models.

Pro tip: Streamline API Testing with Apidog



Looking to simplify your API workflows? **Apidog** acts as an all-in-one solution for creating, managing, and running tests and mock servers. With Apidog, you can:

- Automate critical workflows without juggling multiple tools or writing extensive scripts.
- Maintain smooth CI/CD pipelines.
- Identify bottlenecks to ensure API reliability.

Save time and focus on perfecting your product. Ready to try it? [Give Apidog a spin!](#)

Why DeepSeek R1?

DeepSeek R1, a model comparable to OpenAI's o1 but 95% cheaper, is revolutionizing RAG systems. Developers love it for its:

- **Focused retrieval:** Uses only 3 document chunks per answer.

- **Strict prompting:** Avoids hallucinations with an “I don’t know” response.
- **Local execution:** Eliminates cloud API latency.

What You’ll Need to Build a Local RAG System

1. Ollama

Ollama lets you run models like DeepSeek R1 locally.

- **Download:** [Ollama](#)
- **Setup:** Install and run the following command via your terminal.

```
ollama run deepseek-r1 # For the 7B model (default)
```



Get up and running with large
language models.

Run [Llama 3.3](#), [Phi 4](#), [Mistral](#), [Gemma 2](#), and
other models. Customize and create your own.

Download ↓

Available for macOS, Linux, and
Windows

2. DeepSeek R1 Model Variants

DeepSeek R1 ranges from 1.5B to 671B parameters. Start small with the **1.5B model** for lightweight RAG applications.

```
ollama run deepseek-r1:1.5b
```

Pro Tip: Larger models (e.g., 70B) offer better reasoning but need more RAM.

```
$ ollama run deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8... 0% | 1.2 MB/1.1 GB 304 KB/s 1h1m
```

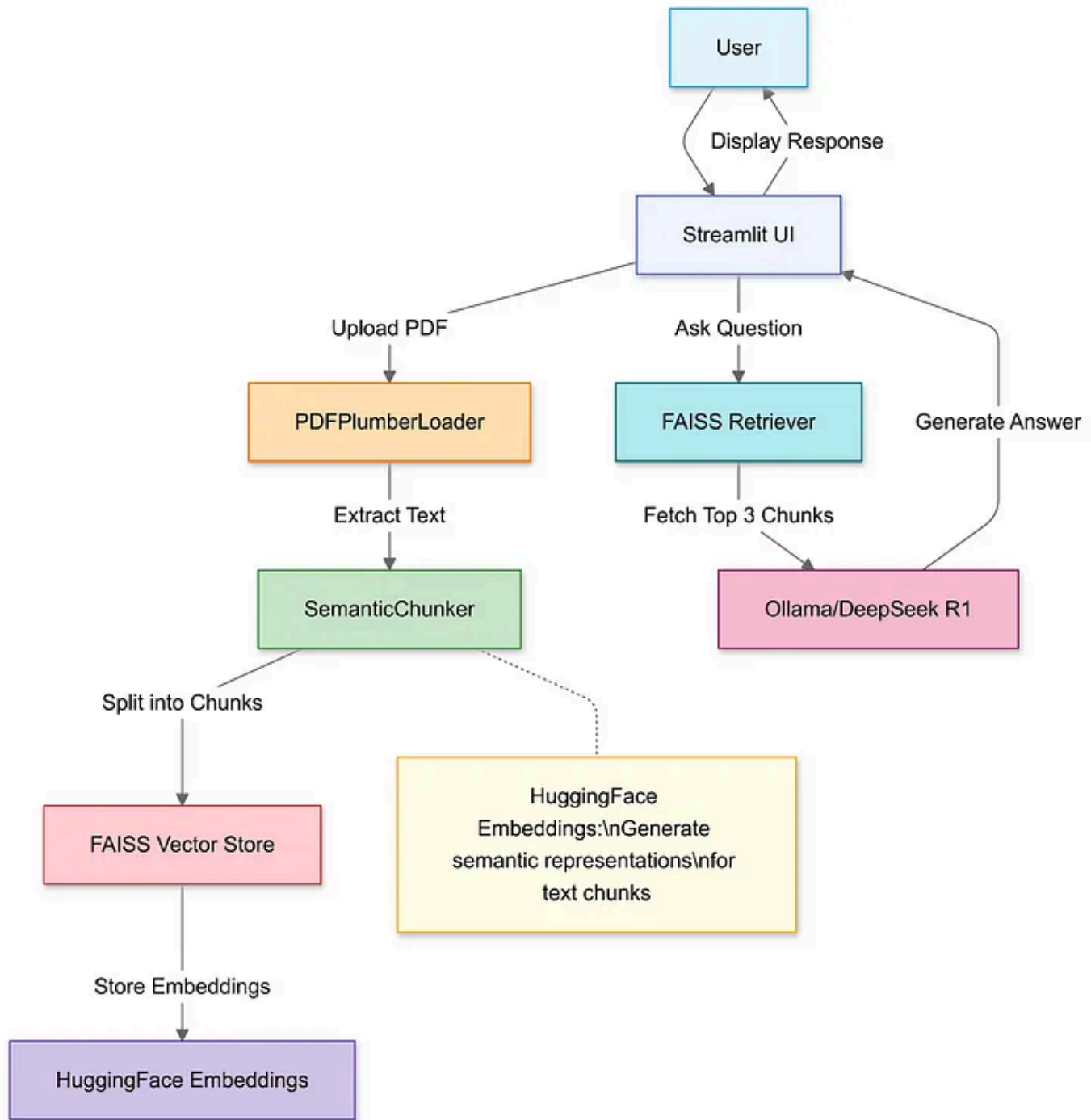
Step-by-Step Guide to Building the RAG Pipeline

Step 1: Import Libraries

We'll use:

- LangChain for document processing and retrieval.
- Streamlit for the user-friendly web interface.

```
import streamlit as st
from langchain_community.document_loaders import PDFPlumberLoader
from langchain_experimental.text_splitter import SemanticChunker
from langchain_community.embeddings import HuggingFaceEmbeddings
from langchain_community.vectorstores import FAISS
from langchain_community.llms import Ollama
```



Step 2: Upload & Process PDFs

Leverage Streamlit's file uploader to select a local PDF. Use `PDFPlumberLoader` to extract text efficiently without manual parsing.

```
# Streamlit file uploader
uploaded_file = st.file_uploader("Upload a PDF file", type="pdf")
```

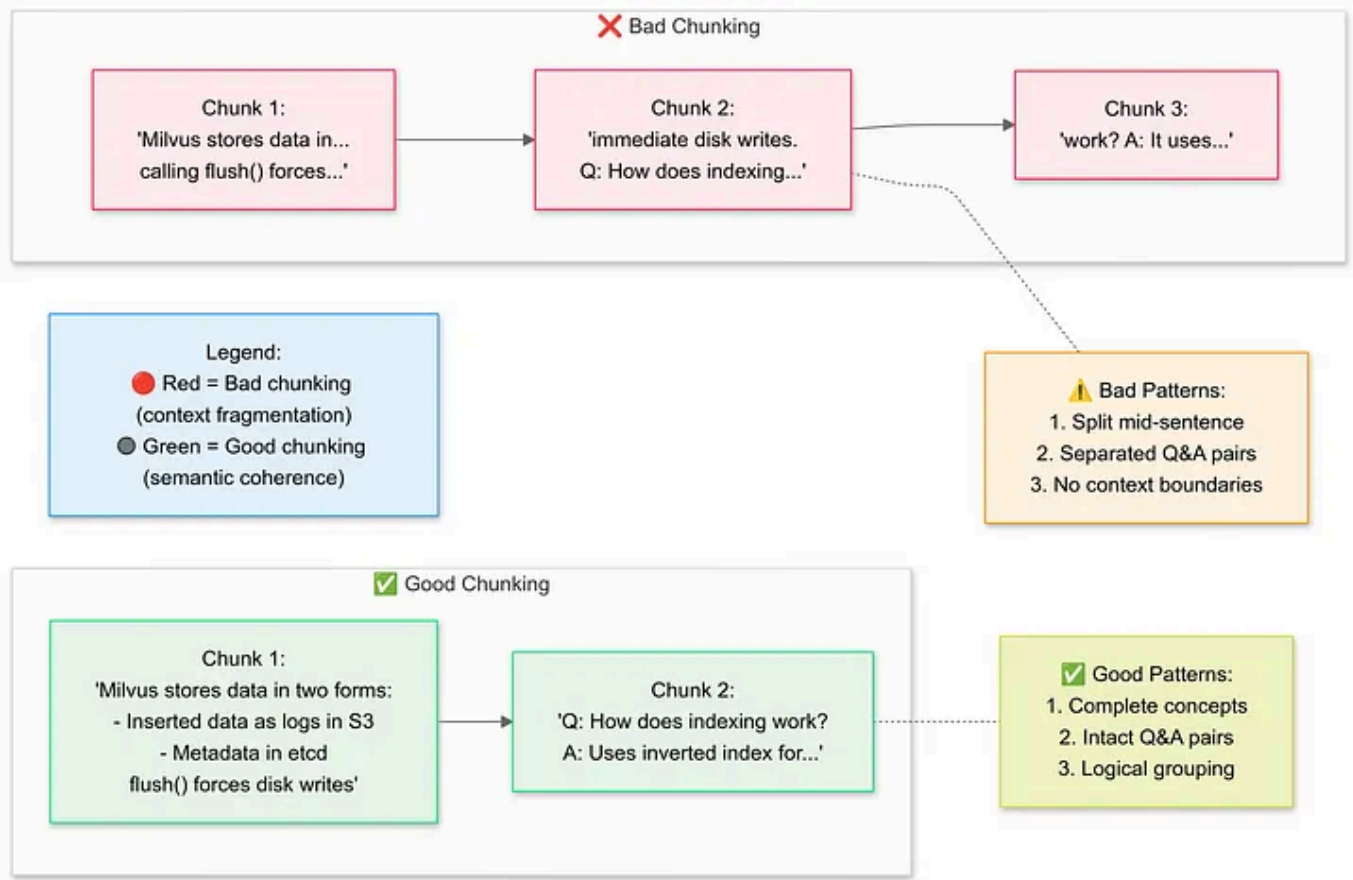
```
if uploaded_file:
    # Save PDF temporarily
    with open("temp.pdf", "wb") as f:
        f.write(uploaded_file.getvalue())

    # Load PDF text
    loader = PDFPlumberLoader("temp.pdf")
    docs = loader.load()
```

Step 3: Chunk Documents Strategically

Leverage Streamlit's file uploader to select a local PDF. Use `PDFPlumberLoader` to extract text efficiently without manual parsing.

```
# Split text into semantic chunks
text_splitter = SemanticChunker(HuggingFaceEmbeddings())
documents = text_splitter.split_documents(docs)
```



Step 4: Create a Searchable Knowledge Base

Generate vector embeddings for the chunks and store them in a FAISS index.

- Embeddings allow fast, contextually relevant searches.

```
# Generate embeddings
embeddings = HuggingFaceEmbeddings()
vector_store = FAISS.from_documents(documents, embeddings)

# Connect retriever
retriever = vector_store.as_retriever(search_kwargs={"k": 3}) # Fetch top 3 chunks
```

Step 5: Configure DeepSeek R1

Set up a RetrievalQA chain using the DeepSeek R1 1.5B model.

- This ensures answers are grounded in the PDF's content rather than relying on the model's training data.

```
llm = Ollama(model="deepseek-r1:1.5b") # Our 1.5B parameter model

# Craft the prompt template
prompt = """
1. Use ONLY the context below.
2. If unsure, say "I don't know".
3. Keep answers under 4 sentences.

Context: {context}

Question: {question}

Answer:
"""
QA_CHAIN_PROMPT = PromptTemplate.from_template(prompt)
```

Step 6: Assemble the RAG Chain

Integrate uploading, chunking, and retrieval into a cohesive pipeline.

- This approach gives the model verified context, enhancing accuracy.

```
# Chain 1: Generate answers
llm_chain = LLMChain(llm=llm, prompt=QA_CHAIN_PROMPT)

# Chain 2: Combine document chunks
document_prompt = PromptTemplate(
    template="Context:\ncontent:{page_content}\nsource:{source}",
    input_variables=["page_content", "source"]
)

# Final RAG pipeline
qa = RetrievalQA(
    combine_documents_chain=StuffDocumentsChain(
        llm_chain=llm_chain,
```



```
        document_prompt=document_prompt
    ),
    retriever=retriever
)
```

Step 7: Launch the Web Interface

Launch the Web Interface

Streamlit enables users to type questions and receive instant answers.

- Queries retrieve matching chunks, feed them to the model, and display results in real-time.

```
# Streamlit UI
user_input = st.text_input("Ask your PDF a question:")

if user_input:
    with st.spinner("Thinking..."):
        response = qa(user_input)["result"]
        st.write(response)
```

Build a RAG System with DeepSeek R1 & Ollama

Drag and drop file here
Limit 200MB per file • PDF

Browse files

Volatility75-Strategy.pdf

How to enter a trade

Response:

<think> Okay, so I need to figure out how to answer the question "How to enter a trade" based on the provided context. Let me start by reading through all the given information carefully.

First, there's some context about drawing lines on M15 and moving to M5 for confirmation. It mentions three scenarios if certain conditions are met or not on M5. Scenario 1 says if all conditions are met on both M15 and M5, mark it as HIGH POTENTIAL. Scenario 2 is when conditions on M15 aren't met on M5; in that case, discard the trade immediately and wait for another signal on M15. Scenario 3 is when conditions on M15 are about to be met on M5; then, wait patiently until the complete signal forms on M5 before entering.

Then there's another section titled "HOW TO ENTER A TRADE." It outlines several steps:

1. All trades should be executed and monitored only on M5.
2. Decide the lot size based on account size, with a guide available elsewhere.
3. Use Stop Orders instead of Instant Execution to prevent entering too soon.
4. When all signals form on all timeframes, switch to M5 and place a STOP ORDER a few PIPS below (for SELL) or above (for BUY) the candlestick.
5. Note that there's always a possibility of a retest before final movement, so placement is cautious.

There's also a strategy overview indicating it's scalping with timeframes from M5 to H1, using Bollinger Bands, RSI, Stochastic Oscillator, and MACD for signals.

Putting this together, the steps to enter a trade primarily involve monitoring signals on M15 first, ensuring they confirm on M30 or H1. If conditions are met, place a Stop Order on M5 with precise pip placement based on the trade type (SELL below, BUY above). Also, lot size is determined by account size.

You can find the complete code here:

<https://gist.github.com/lisakim0/0204d7504d17cefceaf2d37261c1b7d5.js>

The Future of RAG with DeepSeek

DeepSeek R1 is just the beginning. With upcoming features like **self-verification** and **multi-hop reasoning**, future RAG systems could debate and refine their logic autonomously.

Build your own RAG system today and unlock the full potential of document-based AI!

Source of this article: [How to Run Deepseek R1 Locally Using Ollama ?](#)

Deepseek R1

Ollama

Rag

**Written by Sebastian Petrus**

826 Followers · 2 Following

Asist Prof @U of Waterloo, AI/ML, e/acc

Follow

Responses (22)



What are your thoughts?

Respond

**Mario Wolfram**

6 days ago



So, you load a model (that currently creates a lot of fuzz) into Ollama (no engineering degree required here), create a very basic RAG pipeline and wrap it into a streamlit boilerplate. In the end you could not even provide a working github link for the lazy.



27



2 replies

Reply**Ramin Assadollahi**

6 days ago



Great article, unfortunately, the Github link doesn't work.



18



1 reply

[Reply](#)**Michael Loh**

6 days ago



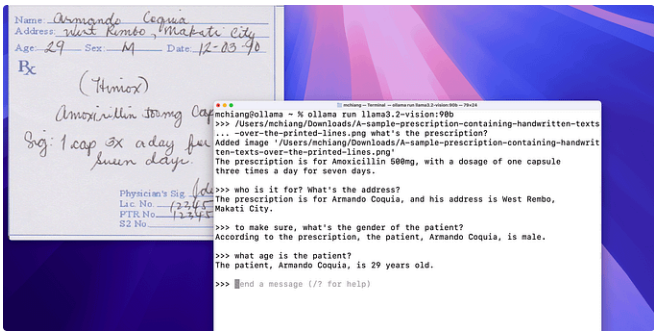
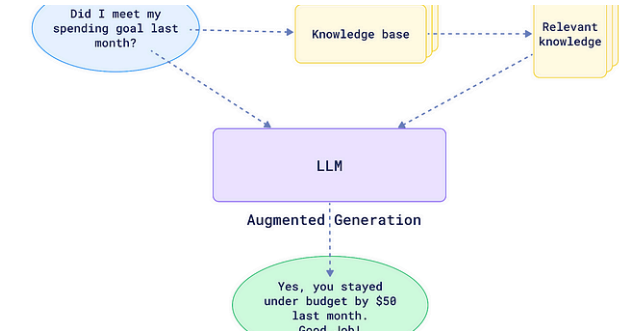
Folks, use the Source of Article link at the end of this article to access the final code.




18

[Reply](#)[See all responses](#)

More from Sebastian Petrus



 Sebastian Petrus

Top 10 RAG Frameworks Github Repos 2024

Retrieval-Augmented Generation (RAG) has emerged as a powerful technique for...

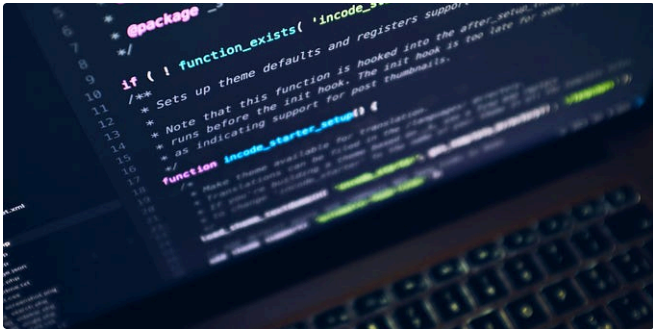
Sep 4, 2024  881  9  

 Sebastian Petrus

Build a Local Ollama OCR Application Using Llama 3.2-Vision

Optical Character Recognition (OCR) has become an essential tool for digitizing printe...

Nov 18, 2024  335  6  

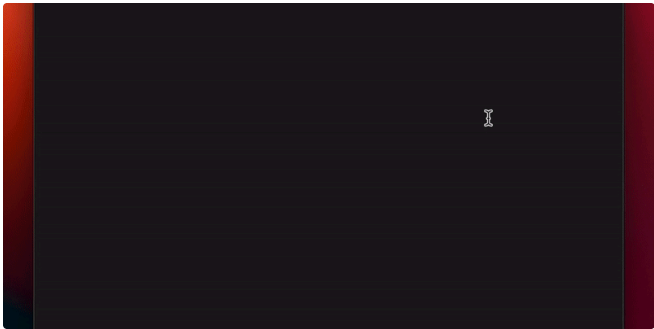



 In Cool Devs by Sebastian Petrus

How to Run Ollama on Google Colab

In the rapidly evolving landscape of artificial intelligence and machine learning, large...

Sep 20, 2024  50  2  



 Sebastian Petrus

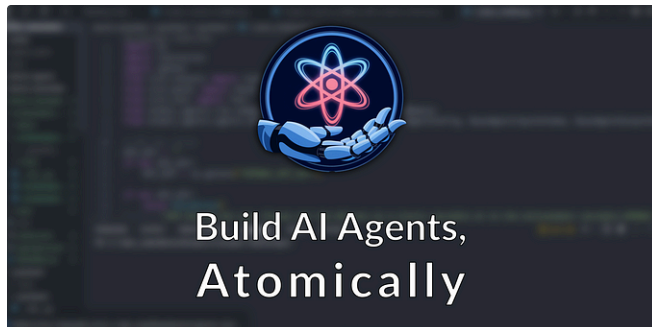
Top 10 Open Source GitHub Copilot Alternatives (2024 Version)

AI-powered coding assistants have become increasingly popular. GitHub Copilot,...

Sep 4, 2024  61  

See all from Sebastian Petrus

Recommended from Medium



 In AI Advances by Kenny Vaneetvelde 

Want to Build AI Agents? Tired of LangChain, CrewAI, AutoGen &...

Frameworks like LangChain, CrewAI, and AutoGen have gained popularity by promisin...

★ Jan 19 🖱 1K 💬 22  ⋮



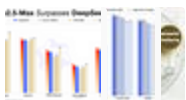
 In Level Up Coding by Dr. Ashish Bamania 

DeepSeek-R1 Beats OpenAI's o1, Revealing All Its Training Secrets...

A deep dive into how DeepSeek-R1 was trained from scratch and how this open-...

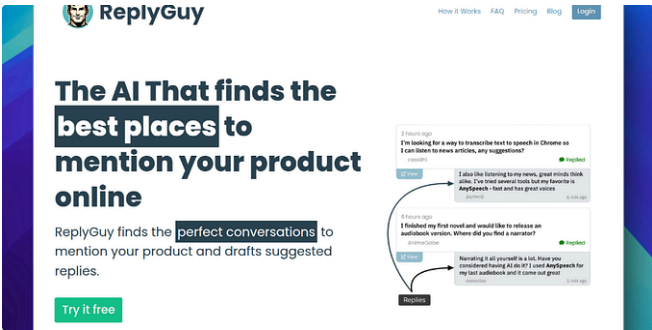
★ 5d ago 🖱 1.1K 💬 28  ⋮

Lists



Natural Language Processing


1908 stories · 1566 saves




 Mohit Vaswani

6 AI Agents That Are So Good, They Feel Illegal

AI agents are the future because they can replace all the manual work with automation...

Jan 11  2.1K  81  

 In Age of Awareness by Cliff Berg

They Know a Collapse Is Coming

The CIO of Goldman Sachs has said that in the next year, companies at the forefront will...

Jan 21  11.3K  565  

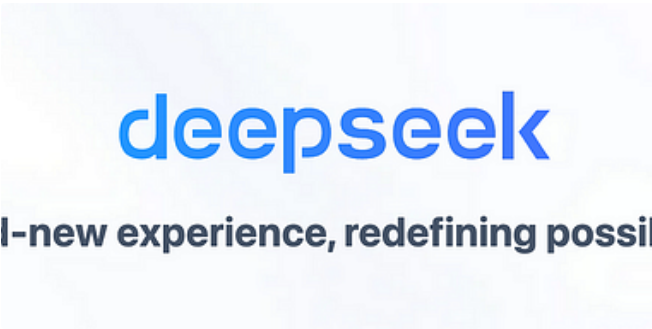


 Pankaj

DeepSeek-R1: A Cutting-Edge Logical Reasoning Model for Loca...

Unlocking AI-Powered Logical Reasoning: How DeepSeek-R1 Revolutionizes Step-by-...


 Jan 22  210  2  



 Abhishek Maheshwarappa

Fine-Tuning DeepSeek LLM: Adapting Open-Source AI for You...

Introduction

Jan 21  110  4  

See more recommendations