



How to Use DeepSeek

 Justin haynes

Thinking ↑

<think>

I need to determine how many times the letter "r" appears in the word "strawberry."

First, I'll write out the word: S-T-R-A-W-B-E-R-R-Y.

Next, I'll go through each letter one by one:

- S has no R.
- T has no R.
- R is present once.
- A has no R.
- W has no R.
- B has no R.
- E has no R.
- R is present twice in "strawberry."
- Y has no R.

Adding them up: 1 (from the first R) + 2 (from the second R) = 3 Rs in total.

</think>



Introducing Today's Project!

In this project, I will demonstrate how to utilize DeepSeek and make the most of this LLM that appears to be performing at the same level as OpenAI - but for a lower price. This project is to determine whether DeepSeek is the new LLM of choice.

Tools and concepts

The services I used were DeepSeek, Ollama, Chatbox, and OpenAI. Key concepts I learned included advanced reasoning, self-hosting, temperature settings and token efficiency. After reviewing DeepSeek vs OpenAI, I prefer to use them both.

Project reflection

This project took me approximately 2 hours to complete, The most challenging part was DeepSeek's overthinking. It was most rewarding to compare token efficiency and utilizing Chatbox for a visual experience with DeepSeek locally.

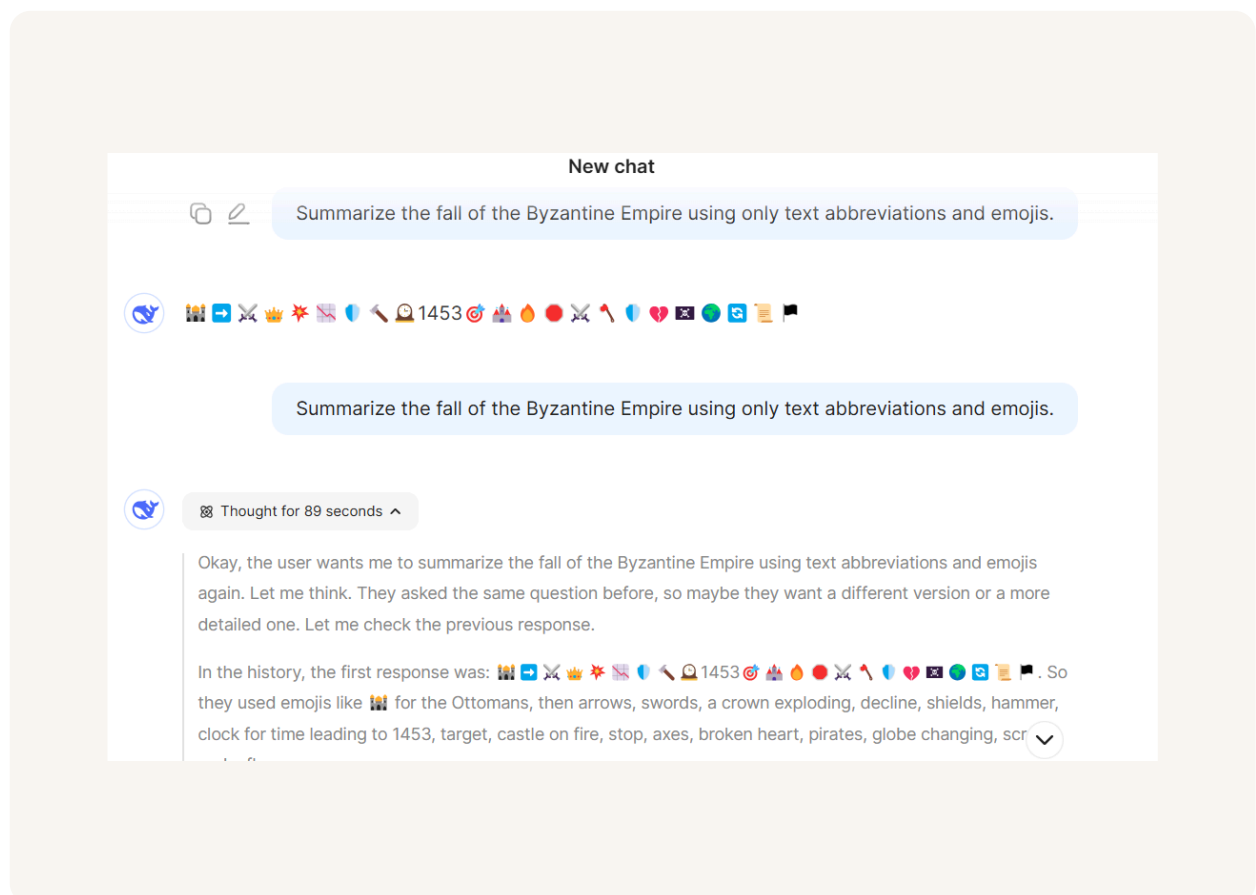
I did this project to have a better understanding of DeepSeek, develop my own opinion, and have a positive comparison to OpenAI. This hands-on experience helped me see both the pros and cons



Exploring DeepSeek

DeepSeek is a company that develops Large Language Models (LLMs). Their R1 model gained attention for its performance that rivals OpenAI's latest o1 model, noting DeepSeek is open source. This means we have access to a world-class LLM for a low price

While I could access DeepSeek over the web app, some of my concerns are privacy (i.e. where the data is stored), internet connection required, and latency (slow response times during high traffic).





Ollama and DeepSeek R1

Ollama is software for using LLMs locally on our computer. It's helpful because it manages the downloading, installing, and updating of an LLM for us, which usually requires a lot of manual work.

You won't be able to find OpenAI models in Ollama because it focuses on open-source models like DeepSeek. OpenAI models are closed systems, so the underlying architecture, codebase, and datasets used to develop them are confidential.

I tested using DeepSeek offline by turning off the Wi-Fi and running commands over the terminal. I observed that I was still receiving a response from DeepSeek. The "think" tags in the terminal indicated DeepSeek's thought process.

```
>>> Can I use DeepSeek while offline?
<think>
Okay, so I'm trying to figure out if I can use DeepSeek while being offline. First off, what exactly is DeepSeek?
From what I remember, it's a search engine that's part of the Baidu ecosystem. So, just like Google has Chrome,
Baidu has something called DeepSeek.

Now, the question is about using it offline. That makes me think about whether you can access it without an
internet connection. Typically, most search engines require an active internet connection to function properly
because they need to fetch data from their servers in real-time. But maybe there's a way to use DeepSeek locally
or with some kind of cached data.

I should consider if DeepSeek has any offline features or modes. Some apps allow you to download data beforehand,
so you can search without the internet. For example, Google has an option where you can pre-cache certain
information and then use it offline. Maybe DeepSeek does something similar.

Another angle is whether there's a desktop application or a mobile app that allows for offline use. If DeepSeek
has an app, perhaps it can store data on your device so you can search without the internet. Alternatively, maybe
through a browser extension or some other tool, you can save information and access it later offline.

I also wonder about the technical feasibility. Search engines rely heavily on real-time data for accuracy and
relevance. Without live data, the results might be outdated or less comprehensive. So, DeepSeek might not support
true offline searches in the traditional sense because their system depends on up-to-the-minute data from the web.

But then again, maybe there are specific use cases where having cached information is beneficial. For instance, if
you're in an area with poor internet connectivity, having some local search functionality could be useful for
basic queries. However, it's unlikely to cover all features of the full search engine.

I should also think about how users typically interact with DeepSeek. If it's integrated into other Baidu services
like Maps or Chrome, maybe there's a way to use those in an offline manner first and then switch back online when
possible. But that doesn't necessarily mean DeepSeek itself supports offline searches.

Another possibility is using a VPN or some sort of local proxy server to access cached data stored on your own
```



DeepSeek R1 Sizes

DeepSeek R1 has different model sizes, which varies in levels of processing needs and accuracy. This is helpful for running DeepSeek locally because we can choose which model best fits our hardware requirements.

The R1 model you choose to run locally depends on your computer's memory and storage space. We chose the 8b model (i.e. 8 billion parameters) because it offers a substantial improvement from the 1.5b model and is still accessible for our computer.

The screenshot shows the 'Settings' window for the DeepSeek R1 application. It features four tabs: 'MODEL' (selected), 'DISPLAY', 'CHAT', and 'ADVANCED'. Under the 'MODEL' tab, the 'Model Provider' is set to 'OLLAMA API'. The 'API Host' is 'http://127.0.0.1:11434' with a 'RESET' button. A light blue informational box states: 'Please ensure that the Remote Ollama Service is able to connect remotely. For more details, refer to [this tutorial](#).' The 'Model' dropdown is set to 'deepseek-r1:1.5b'. Below this, 'Max Message Count in Context' is a slider set to 20. 'Temperature' is a slider set to 1.01, with 'Meticulous' and 'Creative' buttons below it. 'CANCEL' and 'SAVE' buttons are at the bottom right.



Chatbox

To complete my local setup, I installed Chatbox to have a visual interface as I talk to DeepSeek R1. My Chatbox settings use Ollama API as the API model (i.e. the tool that connects us with the desired LLM) and the 1.5b R1 model.

I tested two different R1 model sizes, which were 1.5b and 8b using the prompt 'How many r's are in strawberry?' the results made it clear that 1.5b was the lighter model- it responded with only 2 r's in "strawberry", whereas the 8b model said 3.

Thinking ↑

<think>

I need to determine how many times the letter "r" appears in the word "strawberry."

First, I'll write out the word: S-T-R-A-W-B-E-R-R-Y.

Next, I'll go through each letter one by one:

- S has no R.
- T has no R.
- R is present once.
- A has no R.
- W has no R.
- B has no R.
- E has no R.
- R is present twice in "strawberry."
- Y has no R.

Adding them up: 1 (from the first R) + 2 (from the second R) = 3 Rs in total.

</think>



Temperature Settings

The temperature setting in an LLM determines how creative and 'out of the box' it will be when it generates a response. To see this in action, I increased the temperature and prompted deepSeek to generate a recipe.

I started a third chat with ChatGPT to act as a judge for the responses. This analysis will also help us understand how to spot low vs high temperature generated text.

ChatGPT quickly figured out which piece was generated with a high temperature, because it identified that the high-temperature piece featured more creative ingredients, more steps, and more complexity. The low-temperature response is also simpler.

Reasoned about temperature-driven text variability for 5 seconds ^

Identifying textual style

I'm thinking through which text is from a higher temperature setting. The first is bizarre with weird ingredients and random instructions, indicating a higher temperature.

Based on their style and content, the **first response** is almost certainly the one generated with the **higher temperature** setting. It includes more unusual word choices, unexpected ingredient details, and a generally more "random" or creative style (e.g., references to "pyodam chips," "Alumsesquiodulox," and "crème_frobouuses"), which are common indicators of a higher temperature