Large Language Models from Cohere

Ram N Sangwan

- Getting Started with **Cohere Models**
- Understanding of **Cohere** Models
- Getting Started with **Cohere API**
- Authentication and Access Keys
- The ***Chat*** endpoint.
- Using the Generative AI Playground on OCI/Cohere

Getting Started with **Cohere Models**

What is Cohere?

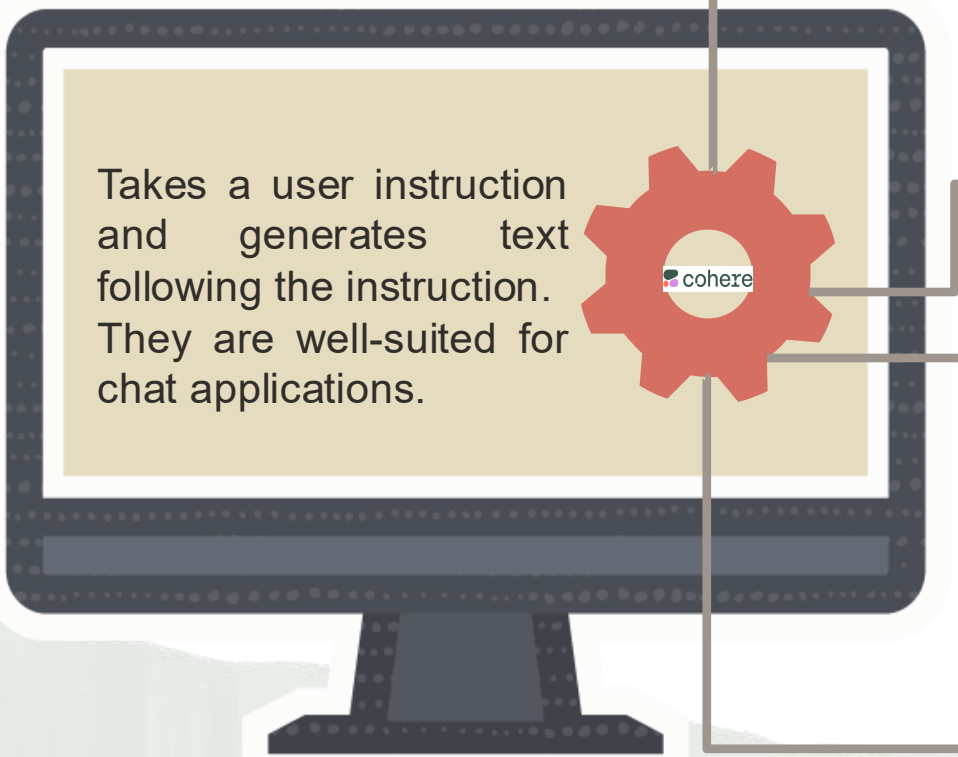
Cohere provides a powerful API for its models that integrates language processing into any system.

Cohere develops large-scale language models and encapsulates them within an intuitive API.

You can tailor these models to suit your use cases.

Cohere provides a range of models that can be trained and tailored to suit specific use cases.

Cohere Baseline Models - Command



command-light - 4096 Tokens

A smaller, faster version of command.

command-light-nightly – 8192 Tokens

Latest, experimental, and unstable version of command-light. Updated regularly, without warning.

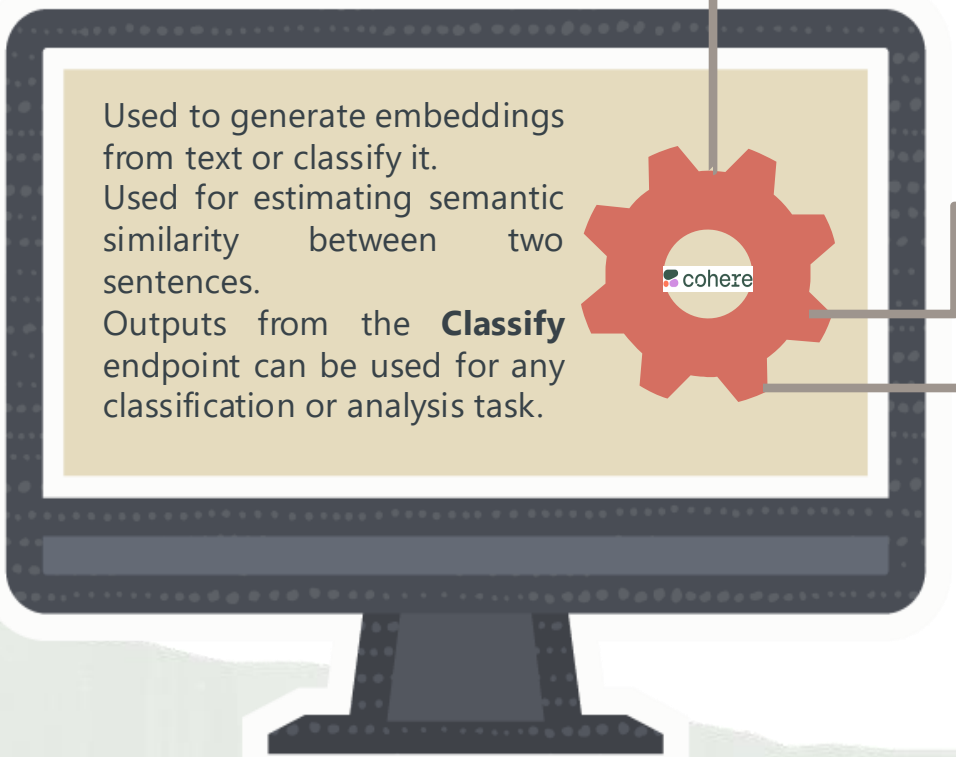
command

An instruction-following conversational model that performs language tasks with high quality, more reliably.

command-nightly

Updating environments, pushing to production

Cohere Baseline Models - **Embed**



embed-english-light-v3.0 - 512 Tokens

A smaller, faster version of embed-english-v3.0

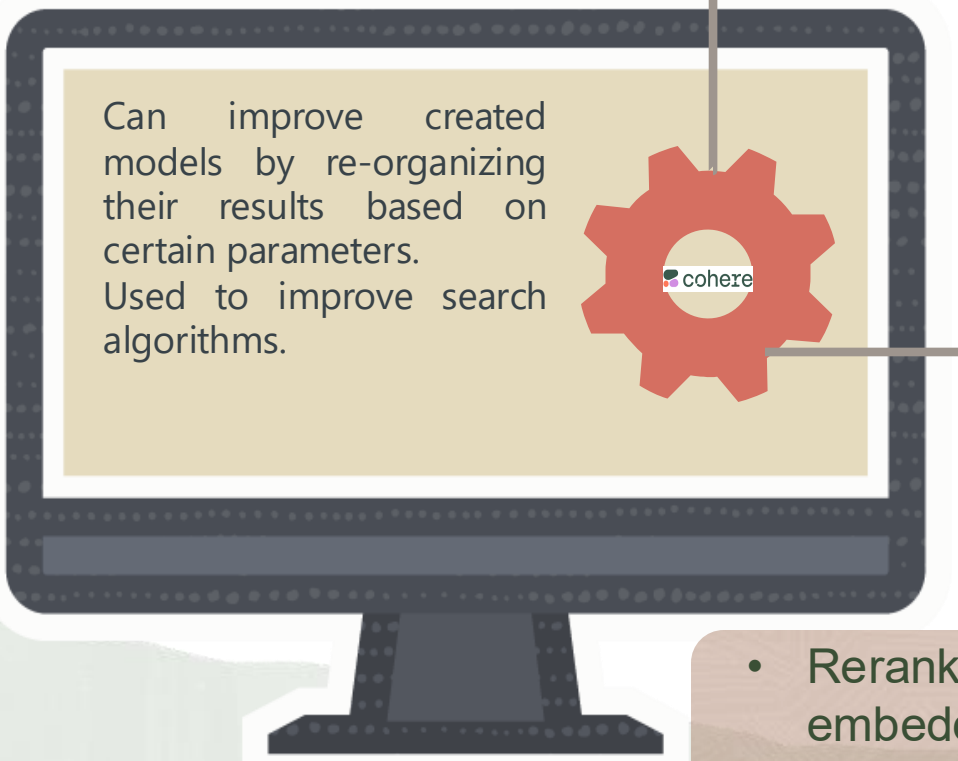
embed-multilingual-v3.0 - 512 Tokens

Provides multilingual classification and embedding support.

embed-multilingual-light-v3.0 - 512 Tokens

A smaller, faster version of embed-multilingual-v3.0.

Cohere Models - Rerank



rerank-english-v2.0

A model that allows for re-ranking English language documents.

rerank-multilingual-v2.0

- A model for documents that are not in English.
- Supports the same languages as embed-multilingual-v3.0.

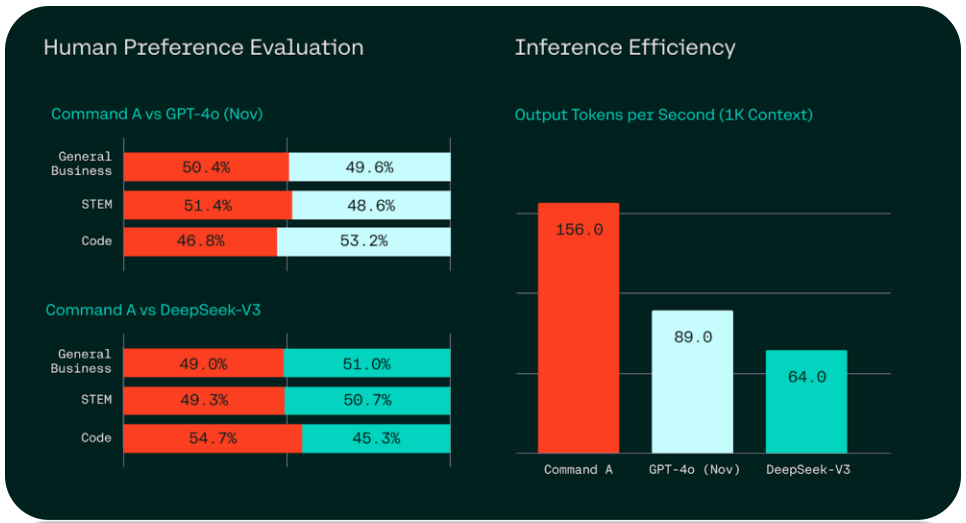
- Rerank not only surpasses the quality of results obtained through embedding-based search but also requires just a single line of code alteration in your application.

Cohere Models – Command-A

Command-A

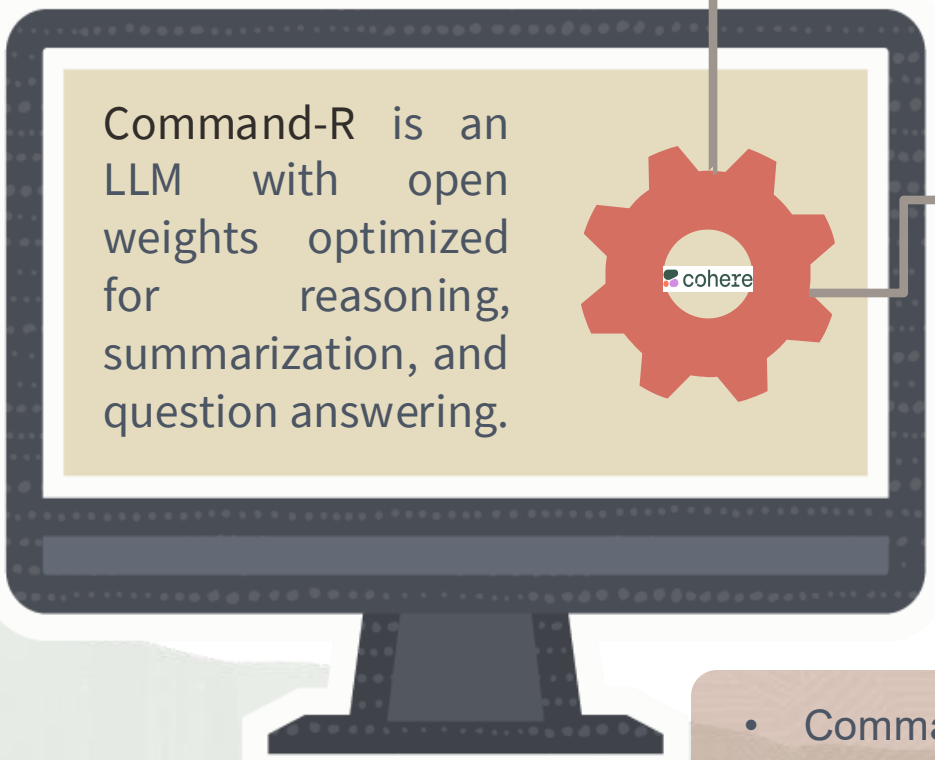
Across a range of standard benchmarks Command A provides strong performance on instruction following, SQL, agentic, and tool tasks.

Command A is on par or better than GPT-4o and DeepSeek-V3 across agentic enterprise tasks, with significantly greater efficiency..



Command A can deliver tokens at a rate of up to 156 tokens/sec which is 1.75x higher than GPT-4o and 2.4x higher than DeepSeek-V3. Private deployments of Command A can be up to 50% cheaper than API-based access

Cohere Models – Command-R



Command-R

With 35 billion parameter, it is highly performant generative model.

Aya

Aya is a multilingual model from Cohere For AI.

Is trained to support 23 languages:

- Arabic, Chinese (simplified, traditional), Czech, Dutch, French, German, Greek, Hindi, Indonesian, Italian, Japanese, Korean, Persian, Polish, Portuguese, Romanian, Russian, Spanish, Turkish, Ukrainian, Vietnamese.

- Command-R has the capability for multilingual generation evaluated in 10 languages and highly performant RAG capabilities.

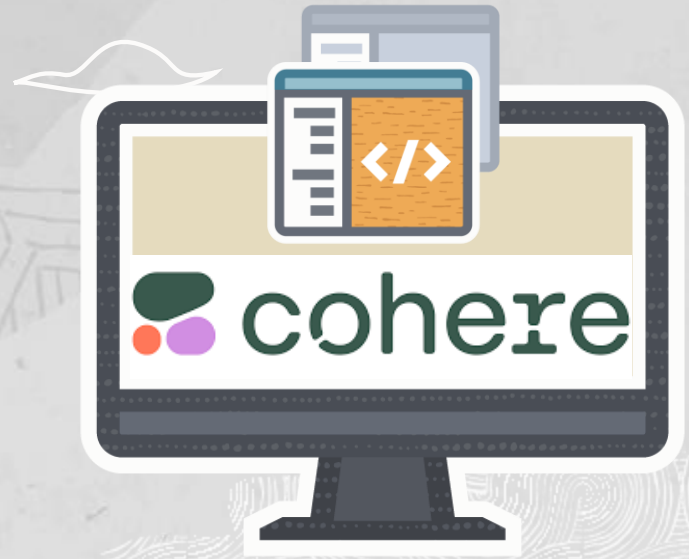
Setting Up Cohere

Register for a Cohere account and get a free to use trial API key.

- There is no credit or time limit associated with a trial key.
- Calls are rate-limited to 10 calls per minute.
- This is typically enough for an experimental project.

Install the Python SDK.

```
pip install cohere
```



Setting Up Cohere

Define the Cohere client with the API key

```
response = co.chat(
    model='command-r-plus-08-2024',
    messages=[
     {"role": "system", "content": system_message},
     {"role": "user",
     "content": "Generate a concise product description for the product: wireless earbuds",
         }
    ],
    max_tokens=2000,
    temperature=temp)
 print(response.message.content[0].text)
```

We defined a some parameters.

- model — We selected command.
- max_tokens — The maximum number of tokens to be generated. One word is about three tokens.

Cohere API Endpoints- Chat

https://api.cohere.com/v2/chat

Generates a text response to a user message.

Create Prompt

Store the message you want to send into a variable

```
messages=[{"role": "user", "content": "hello world!"}]
```

Define the Model Settings

model: command, command-light, command-nightly, and command-light-nightly.
temperature: Controls the randomness of the output.

Generate the Response

```
import cohere
co = cohere.ClientV2()
response = co.chat(
    model="command-r-plus-08-2024",
    messages=[{"role": "user", "content": "hello world!"}],
)
print(response)
```

The endpoint has a number of settings you can use to control the kind of output it generates.

Old Chat API Example – V1

```
import cohere
co = cohere.Client('<<apiKey>>')
response = co.chat(
  chat_history=[
    {"role": "USER", "message": "Who discovered gravity?"},
    {"role": "CHATBOT", "message": "The man who is widely
credited with discovering gravity is Sir Isaac Newton"}
  ],
  message="What year was he born?",
  # perform web search before answering the question. You can
also use your own custom connector.
  connectors=[{"id": "web-search"}]
)
print(response)
```

A list of previous messages between the user and the model, meant to give the model conversational context for responding to the user's message.

One of CHATBOT|USER to identify who the message is coming from.

Text input for the model to respond to.

Embed

https://api.cohere.ai/v1/embed

- Returns text embeddings.
- An embedding is a list of floating point numbers that captures semantic information about the text that it represents.
- Embeddings can be used to create text classifiers as well as empower semantic search.

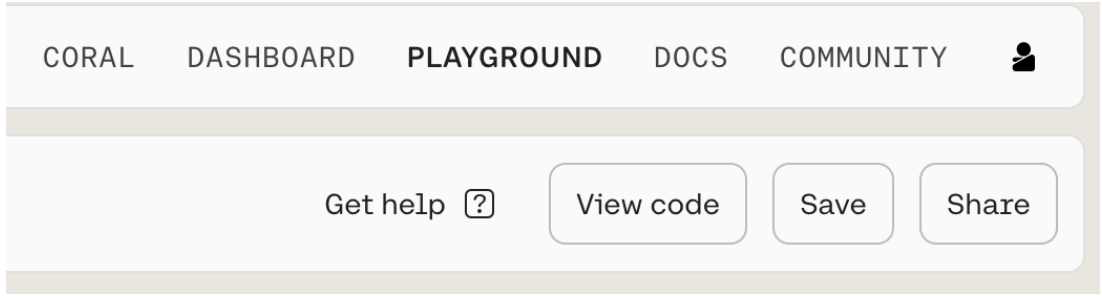
```
import cohere
co = cohere.Client('<<apiKey>>')

response = co.embed(
  texts=['hello', 'goodbye'],
  model='embed-english-v3.0',
  input_type='classification'
)
print(response)
```


```
{
  "response_type": "embeddings_floats",
  "id": "string",
  "embeddings": [
    [
      0
    ]
  ],
  "texts": [
    "string"
  ],
  "meta": {
    "api_version": {
      "version": "string",
      "is_deprecated": true,
      "is_experimental": true
    },
    "billed_units": {
      "input_tokens": 0,
      "output_tokens": 0,
      "search_units": 0,
      "classifications": 0
    },
    "warnings": [
      "string"
    ]
  }
}
```

Cohere Playground

- A visual interface for users to test Cohere’s LLMs without writing a single line of code.



Why Use Cohere Playground?

Introduction to AI Technology

- Serves as a fantastic introduction to the incredible capabilities of AI technology.
- By playing with different AI models, you can experience first-hand their unique strengths and discover how AI can benefit you.

Research & Learning

- A haven for AI research and learning.
- By interacting with the pre-existing AI models, you can gain insights into how these models respond to various prompts and how they generate human-like text.
- Whether you're a student trying to understand the nuances of AI for your thesis or a business professional looking to leverage AI for your operations, the Playground is an invaluable resource.

Innovation & Creativity

Innovation

- Allows you to push the boundaries of what's possible with AI, to imagine and realize new applications, and to contribute to the AI revolution in your own unique way.

Be more Creative

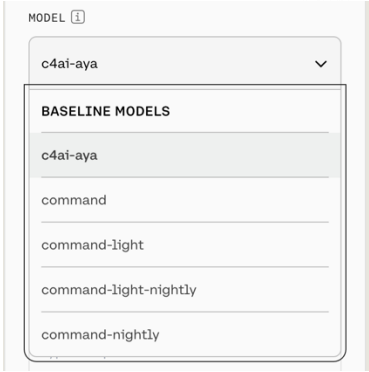
- Whether you're using AI to automate repetitive tasks, generate novel content, or understand complex data, the Playground is the canvas where you bring your AI-powered ideas to life.

Getting Started with Cohere Playground

After registration, head over to the Cohere Playground

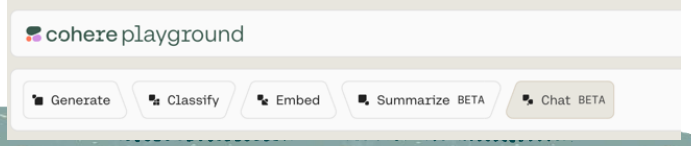
Choose your Parameters

Try tinkering with different temperature and token-picking settings to alter the model's output behaviour.



On the top you will see the 5 tabs:

- Generate,
- Classify,
- Embed,
- Summarize and
- Chat.



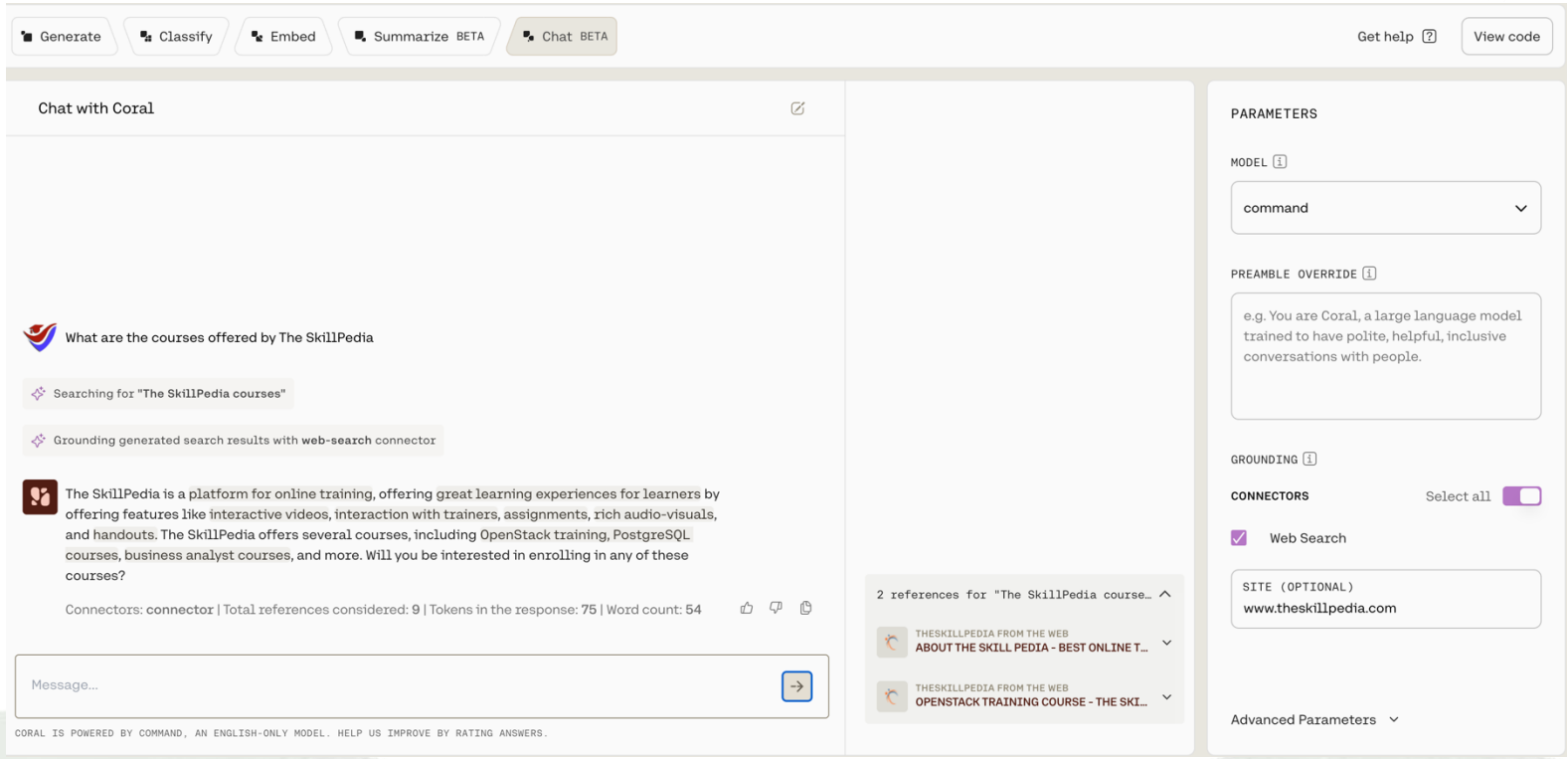
Pickup your Model

Cohere Playground offers a variety of models, each with its own set of strengths.

For example, **command** is a popular choice for its superior text generation abilities.

Chat API Example

You can search a specific website and ask questions about it.



Thank You