

Day43_Project_Text_Generation_Using_Generative_AI

July 12, 2025

Text Generation Using Google Generative AI (Gemini)

This project demonstrates how to build a simple **text generation system** using **Google's Generative AI (Gemini 2.5 Flash model)**.

We connect to Gemini using Python, send a prompt, and receive an AI-generated response.

What You'll Learn

- What is Generative AI and how it works
- How to use `google-generativeai` Python SDK
- How to send text prompts and receive creative or informative replies
- How to print, style, and format model outputs using `IPython.display`

Technologies Used

- Python
- Google Generative AI (`google-generativeai`)
- Gemini 2.5 Flash model
- IPython for Markdown output

Setup Instructions

```
pip install -U google-generativeai
```

Make sure to get your API key from:

- Link : <https://makersuite.google.com/app/apikey>
- Enable Generative Language API in Google Cloud.

1 Install Required Library

Install Google Generative AI SDK (only needs to be run once)

```
[2]: pip install -U google-generativeai
```

```
Collecting google-generativeai
  Using cached google_generativeai-0.8.5-py3-none-any.whl.metadata (3.9 kB)
Collecting google-ai-generativelanguage==0.6.15 (from google-generativeai)
  Using cached google_ai_generativelanguage-0.6.15-py3-none-any.whl.metadata (5.7 kB)
Collecting google-api-core (from google-generativeai)
  Using cached google_api_core-2.25.1-py3-none-any.whl.metadata (3.0 kB)
Collecting google-api-python-client (from google-generativeai)
```

Using cached google_api_python_client-2.176.0-py3-none-any.whl.metadata (7.0 kB)

Collecting google-auth>=2.15.0 (from google-generativeai)

Using cached google_auth-2.40.3-py2.py3-none-any.whl.metadata (6.2 kB)

Requirement already satisfied: protobuf in c:\users\lenovo\anaconda3\lib\site-packages (from google-generativeai) (5.29.3)

Requirement already satisfied: pydantic in c:\users\lenovo\anaconda3\lib\site-packages (from google-generativeai) (2.10.3)

Requirement already satisfied: tqdm in c:\users\lenovo\anaconda3\lib\site-packages (from google-generativeai) (4.67.1)

Requirement already satisfied: typing-extensions in c:\users\lenovo\anaconda3\lib\site-packages (from google-generativeai) (4.12.2)

Collecting proto-plus<2.0.0dev,>=1.22.3 (from google-ai-generativelanguage==0.6.15->google-generativeai)

Using cached proto_plus-1.26.1-py3-none-any.whl.metadata (2.2 kB)

Collecting googleapis-common-protos<2.0.0,>=1.56.2 (from google-api-core->google-generativeai)

Using cached googleapis_common_protos-1.70.0-py3-none-any.whl.metadata (9.3 kB)

Requirement already satisfied: requests<3.0.0,>=2.18.0 in c:\users\lenovo\anaconda3\lib\site-packages (from google-api-core->google-generativeai) (2.32.3)

Collecting grpcio<2.0.0,>=1.33.2 (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,>=1.34.1->google-ai-generativelanguage==0.6.15->google-generativeai)

Using cached grpcio-1.73.1-cp313-cp313-win_amd64.whl.metadata (4.0 kB)

Collecting grpcio-status<2.0.0,>=1.33.2 (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,>=1.34.1->google-ai-generativelanguage==0.6.15->google-generativeai)

Using cached grpcio_status-1.73.1-py3-none-any.whl.metadata (1.1 kB)

Requirement already satisfied: cachetools<6.0,>=2.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from google-auth>=2.15.0->google-generativeai) (5.5.1)

Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\lenovo\anaconda3\lib\site-packages (from google-auth>=2.15.0->google-generativeai) (0.2.8)

Collecting rsa<5,>=3.1.4 (from google-auth>=2.15.0->google-generativeai)

Using cached rsa-4.9.1-py3-none-any.whl.metadata (5.6 kB)

INFO: pip is looking at multiple versions of grpcio-status to determine which version is compatible with other requirements. This could take a while.

Collecting grpcio-status<2.0.0,>=1.33.2 (from google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,>=1.34.1->google-ai-generativelanguage==0.6.15->google-generativeai)

Using cached grpcio_status-1.73.0-py3-none-any.whl.metadata (1.1 kB)

Using cached grpcio_status-1.72.2-py3-none-any.whl.metadata (1.1 kB)

Using cached grpcio_status-1.72.1-py3-none-any.whl.metadata (1.1 kB)

Using cached grpcio_status-1.71.2-py3-none-any.whl.metadata (1.1 kB)

Requirement already satisfied: charset-normalizer<4,>=2 in

```

c:\users\lenovo\anaconda3\lib\site-packages (from
requests<3.0.0,>=2.18.0->google-api-core->google-generativeai) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\lenovo\anaconda3\lib\site-packages (from
requests<3.0.0,>=2.18.0->google-api-core->google-generativeai) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\lenovo\anaconda3\lib\site-packages (from
requests<3.0.0,>=2.18.0->google-api-core->google-generativeai) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\lenovo\anaconda3\lib\site-packages (from
requests<3.0.0,>=2.18.0->google-api-core->google-generativeai) (2025.4.26)
Requirement already satisfied: pyasn1>=0.1.3 in
c:\users\lenovo\anaconda3\lib\site-packages (from rsa<5,>=3.1.4->google-
auth>=2.15.0->google-generativeai) (0.4.8)
Collecting httplib2<1.0.0,>=0.19.0 (from google-api-python-client->google-
generativeai)
Using cached httplib2-0.22.0-py3-none-any.whl.metadata (2.6 kB)
Collecting google-auth-httplib2<1.0.0,>=0.2.0 (from google-api-python-
client->google-generativeai)
Using cached google_auth_httplib2-0.2.0-py2.py3-none-any.whl.metadata (2.2 kB)
Collecting uritemplate<5,>=3.0.1 (from google-api-python-client->google-
generativeai)
Using cached uritemplate-4.2.0-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied:
pyparsing!=3.0.0,!3.0.1,!3.0.2,!3.0.3,<4,>=2.4.2 in
c:\users\lenovo\anaconda3\lib\site-packages (from
httplib2<1.0.0,>=0.19.0->google-api-python-client->google-generativeai) (3.2.0)
Requirement already satisfied: annotated-types>=0.6.0 in
c:\users\lenovo\anaconda3\lib\site-packages (from pydantic->google-generativeai)
(0.6.0)
Requirement already satisfied: pydantic-core==2.27.1 in
c:\users\lenovo\anaconda3\lib\site-packages (from pydantic->google-generativeai)
(2.27.1)
Requirement already satisfied: colorama in c:\users\lenovo\anaconda3\lib\site-
packages (from tqdm->google-generativeai) (0.4.6)
Using cached google_generativeai-0.8.5-py3-none-any.whl (155 kB)
Using cached google_ai_generativelanguage-0.6.15-py3-none-any.whl (1.3 MB)
Using cached google_api_core-2.25.1-py3-none-any.whl (160 kB)
Using cached google_auth-2.40.3-py2.py3-none-any.whl (216 kB)
Using cached googleapis_common_protos-1.70.0-py3-none-any.whl (294 kB)
Using cached grpcio-1.73.1-cp313-cp313-win_amd64.whl (4.3 MB)
Using cached grpcio_status-1.71.2-py3-none-any.whl (14 kB)
Using cached proto_plus-1.26.1-py3-none-any.whl (50 kB)
Using cached rsa-4.9.1-py3-none-any.whl (34 kB)
Using cached google_api_python_client-2.176.0-py3-none-any.whl (13.7 MB)
Using cached google_auth_httplib2-0.2.0-py2.py3-none-any.whl (9.3 kB)
Using cached httplib2-0.22.0-py3-none-any.whl (96 kB)
Using cached uritemplate-4.2.0-py3-none-any.whl (11 kB)

```

Installing collected packages: writemplate, rsa, proto-plus, httpplib2, grpcio, googleapis-common-protos, grpcio-status, google-auth, google-auth-httpplib2, google-api-core, google-api-python-client, google-ai-generativelanguage, google-generativeai

```
----- 1/13 [rsa]
----- 2/13 [proto-plus]
----- 4/13 [grpcio]
----- 4/13 [grpcio]
----- 4/13 [grpcio]
----- 4/13 [grpcio]
----- 4/13 [grpcio]
----- 5/13 [googleapis-common-protos]
----- 5/13 [googleapis-common-protos]
----- 5/13 [googleapis-common-protos]
----- 6/13 [grpcio-status]
----- 7/13 [google-auth]
----- 7/13 [google-auth]
----- 7/13 [google-auth]
----- 7/13 [google-auth]
----- 9/13 [google-api-core]
----- 9/13 [google-api-core]
----- 9/13 [google-api-core]
----- 9/13 [google-api-core]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 10/13 [google-api-python-client]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
```

```

----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 11/13 [google-ai-generativelanguage]
----- 12/13 [google-generativeai]
----- 12/13 [google-generativeai]
----- 12/13 [google-generativeai]
----- 12/13 [google-generativeai]
----- 12/13 [google-generativeai]
----- 12/13 [google-generativeai]
----- 13/13 [google-generativeai]

```

Successfully installed google-ai-generativelanguage-0.6.15 google-api-core-2.25.1 google-api-python-client-2.176.0 google-auth-2.40.3 google-auth-httpplib2-0.2.0 google-generativeai-0.8.5 googleapis-common-protos-1.70.0 grpcio-1.73.1 grpcio-status-1.71.2 httpplib2-0.22.0 proto-plus-1.26.1 rsa-4.9.1 uritemplate-4.2.0

Note: you may need to restart the kernel to use updated packages.

2 Import the Required Libraries

```
[3]: import google.generativeai as genai
```

3 Configure Your API Key

- Replace “Your Key” with your actual key

from <https://makersuite.google.com/app/apikey>

Make sure:

- You’ve enabled the Generative Language API in your Google Cloud account.
- Don’t share your key publicly.

```
[4]: genai.configure(api_key="Your Key")
```

4 Load the Gemini Text Generation Model

```
[5]: model = genai.GenerativeModel("gemini-2.5-flash-preview-04-17")
```

5 Generate a Response to a Smart Prompt

Asking about AI, Generative AI, and Agentic AI in simple terms

5.1 Generate a text response from a prompt

```
[6]: response = model.generate_content("Hello! How are you doing today?")
print(response.text)
```

Hello! Thank you for asking.

As an AI, I don't experience feelings or physical states in the way humans do, but I'm functioning perfectly and ready to help you.

How are **you** doing today? What can I do for you?

5.2 Generate a text response from a prompt

```
[7]: response = model.generate_content(
    "Explain in simple words:\n"
    "1. What is Artificial Intelligence (AI)?\n"
    "2. What is Generative AI?\n"
    "3. What is Agentic AI?\n"
    "4. How are they different from each other?\n"
    "Give short, beginner-friendly answers."
)

print(response.text)
```

Okay, let's break these down in simple terms.

1. **What is Artificial Intelligence (AI)?**
* **Simple answer:** AI is about making computers think or act in smart ways, like humans do, especially when it comes to learning, solving problems, or making decisions. It's trying to give computers "brains" to do tasks that usually need human intelligence.
2. **What is Generative AI?**
* **Simple answer:** Generative AI is a *type* of AI that can **create new things**. Think of it like a computer artist or writer. It learns from lots of examples and then can generate brand new text, images, music, code, or other content that didn't exist before.
3. **What is Agentic AI?**

* **Simple answer:** Agentic AI is a *type* of AI that can **take action** and make its own decisions to reach a specific goal you give it. Instead of just answering a question, it can break down a task, figure out the steps needed, use tools (maybe even other AIs), and work towards completing that task on its own, like a smart assistant trying to finish a project for you.

4. **How are they different from each other?**

* **AI** is the big overall idea – making computers smart.

* **Generative AI** is one *specific way* AI is smart: by **creating** new content (like writing or drawing).

* **Agentic AI** is another *specific way* AI is smart: by **doing tasks** and **reaching goals** through planning and taking action. It's about the AI **acting** in the world to achieve something, which might even involve **using** generative AI along the way!

6 Format the Output Nicely in Markdown

```
[8]: import textwrap
from IPython.display import Markdown

def to_markdown(text):
    text = text.replace('-', ' *')
    return Markdown(textwrap.indent(text, '> ', predicate=lambda _: True))

to_markdown(response.text)
```

[8]: Okay, let's break these down in simple terms.

1. What is Artificial Intelligence (AI)?

- **Simple answer:** AI is about making computers think or act in smart ways, like humans do, especially when it comes to learning, solving problems, or making decisions. It's trying to give computers “brains” to do tasks that usually need human intelligence.

2. What is Generative AI?

- **Simple answer:** Generative AI is a *type* of AI that can **create new things**. Think of it like a computer artist or writer. It learns from lots of examples and then can generate brand new text, images, music, code, or other content that didn't exist before.

3. What is Agentic AI?

- **Simple answer:** Agentic AI is a *type* of AI that can **take action and make its own decisions** to reach a specific goal you give it. Instead of just answering a question, it can break down a task, figure out the steps needed, use tools (maybe even other AIs), and work towards completing that task on its own, like a smart assistant trying to finish a project for you.

4. How are they different from each other?

- **AI** is the big overall idea – making computers smart.
- **Generative AI** is one *specific way* AI is smart: by **creating** new content (like writing or drawing).

- **Agentic AI** is another *specific way* AI is smart: by **doing tasks** and *reaching goals* through planning and taking action. It's about the AI *acting* in the world to achieve something, which might even involve *using* generative AI along the way!

7 Conclusion

- We successfully built a text generation app using Gemini 2.5 Flash.
- Learned how to prompt Gemini using Python and read AI-generated content.
- This forms the basis for more advanced chatbots, agents, or creative tools.

Next, we'll move to multi-modal input — where Gemini will generate content from **images**, not just text!