

monitoring. Security. Databases. Buy third-party software via AWS.



Hugging Face

<https://huggingface.co>

## Hugging Face – The AI community building the future.

The AI community building the future. The platform where the machine learning community collaborates on models, datasets, and applications. Explore AI Apps.



### Kolors Virtual Try-On in the Wild

Upload a person image and a garment image to see how the ...



### Spaces

Image Generation · Video Generation · Text Generation ...



### Log In

We're on a journey to advance and democratize artificial ...



### Models


Explore machine learning models.



### Learn

We're on a journey to advance and democratize artificial ...

AI Mode **All** Videos Images Shopping News Short videos More Tools

 **Google Colab**  
https://colab.research.google.com

## Welcome To Colab - Colab - Google

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab ...

→ **Colab notebook**  
Sign in.

→ **Jupyter.ipynb**  
When a notebook contains more cells, we can click "Kernel ...

→ **Run in Google Colab**  
Colab Specifics ... Colab is a virtual machine you can access ...

→ **Pro**  
Colab Pro+ ... Limited time offer of an additional 100 compute units ...

→ **Explore the Gemini API**  
Colab notebooks allow you to combine executable code and ...

[More results from google.com »](#)

## Videos :



**How to Use GOOGLE COLAB | Google Colab Tutorials for ...**

YouTube · GeeksforGeeks  
22 Dec 2023

Commands + Code + Text Run all Cannot save changes

RAM Disk

```
[2]
✓ 3m
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
import PyPDF2
import io

# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response

def extract_text_from_pdf(pdf_file):
    if pdf_file is None:
        return ""
```



[1]

✓ 13s

!pip install transformers torch gradio PyPDF2 -q

Run cell (⌘/Ctrl+Enter)  
cell executed since last change 232.6/232.6 kB 11.8 MB/s eta 0:00:00

[ ]

executed by Harshitha  
11:03 AM (0 minutes ago)  
executed in 13.529s

gr

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import PyPDF2
import io

# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
```

IBM-project Public

Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file

Add file

Code

About

Harshitha0142	Add files via upload	87b2336 · now	4 Commits
README.md	Initial commit	last week	
SmartAi.mp4	Add files via upload	1 minute ago	
Sustainable Smart City Assistant Project Do...	Add files via upload	now	
smartai.py	Add files via upload	last week	

README

IBM-project

No description, website, or topics provided.

- Readme
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Python 100.0%

Suggested workflows

Based on your tech stack

colab.research.google.com

RAM Disk

Commands + Code + Text ▶ Run all Cannot save changes

[2] 3m

```
)
summarize_btn = gr.Button("Summarize Policy")

with gr.Column():
    summary_output = gr.Textbox(label="Policy Summary & Key Points", lines=20)

    summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)

app.launch(share=True)
```

To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret in your Google Cloud Platform project. You will be able to reuse this secret in all of your notebooks. Please note that authentication is recommended but still optional to access public models or datasets.

warnings.warn(
tokenizer\_config.json: 8.88k/? [00:00<00:00, 185kB/s]
vocab.json: 777k/? [00:00<00:00, 8.57MB/s]
merges.txt: 442k/? [00:00<00:00, 5.67MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 33.6MB/s]
added\_tokens.json: 100% 87.0/87.0 [00:00<00:00, 2.29kB/s]
special\_tokens\_map.json: 100% 701/701 [00:00<00:00, 12.0kB/s]
config.json: 100% 786/786 [00:00<00:00, 73.3kB/s]
`torch\_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.21MB/s]
Fetching 2 files: 100% 2/2 [02:34<00:00, 154.46s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:33<00:00, 48.1MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 19.5MB/s]
Loading checkpoint shards: 100% 2/2 [00:20<00:00, 8.40s/it]
generation\_config.json: 100% 137/137 [00:00<00:00, 9.84kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
\* Running on public URL: <https://837cae8elec67ac0bd.gradio.live>
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to

Variables Terminal

11:07 AM T4 (Python 3)