Here is a complete **Python script** that summarizes lengthy articles using **Natural Language Processing (NLP)** techniques. This example uses **transformers** by Hugging Face and a pre-trained model like facebook/bart-large-cnn for extractive and abstractive summarization.

**✅ Requirements**

Install the required libraries:

bash

CopyEdit

pip install transformers torch

**🧠 Python Script for Article Summarization**

python

CopyEdit

from transformers import pipeline

# Initialize summarization pipeline using a pre-trained model

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

# Sample lengthy article (you can replace this with any long text)

article = """

India is a country in South Asia. It is the seventh-largest country by land area,

the most populous country as of 2023, and the most populous democracy in the world.

Bounded by the Indian Ocean on the south, the Arabian Sea on the southwest,

and the Bay of Bengal on the southeast, it shares land borders with Pakistan to the northwest;

China, Nepal, and Bhutan to the north; and Bangladesh and Myanmar to the east.

In the Indian Ocean, India is in the vicinity of Sri Lanka and the Maldives;

its Andaman and Nicobar Islands share a maritime border with Thailand, Myanmar, and Indonesia.

Modern humans arrived on the Indian subcontinent from Africa no later than 55,000 years ago.

Their long occupation, initially in varying forms of isolation as hunter-gatherers,

has made the region highly diverse, second only to Africa in human genetic diversity.

Settled life emerged on the subcontinent in the western margins of the Indus River basin 9,000 years ago,

evolving gradually into the Indus Valley Civilisation of the third millennium BCE.

By 1200 BCE, an archaic form of Sanskrit, an Indo-European language, had diffused into India

from the northwest, unfolding as the language of the Vedas, and recording the dawning of Hinduism in India.

The Dravidian languages of India were supplanted in the northern and western regions.

By 400 BCE, stratification and exclusion by caste had emerged within Hinduism,

and Buddhism and Jainism had arisen, proclaiming social orders unlinked to heredity.

Early political consolidations gave rise to the Maurya and Gupta Empires based in the Ganges Basin.

Their collective era was suffused with wide-ranging creativity, but also marked by the declining status of women,

and the incorporation of untouchability into an organized system of belief.

In South India, the Middle kingdoms exported Dravidian-languages and culture to Southeast Asia.

"""

# Summarize the article

summary = summarizer(article, max\_length=130, min\_length=30, do\_sample=False)

# Display the results

print("Original Length:", len(article.split()))

print("Original Article:\n", article)

print("\n---\n")

print("Summary:\n", summary[0]['summary\_text'])

**🧪 Sample Output:**

text

CopyEdit

Original Length: 310

Summary:

India is a populous democracy located in South Asia, surrounded by several countries and oceans.

Modern humans arrived around 55,000 years ago, leading to rich genetic diversity.

Ancient civilizations such as the Indus Valley and empires like Maurya and Gupta shaped its history.

Hinduism, Buddhism, and Jainism originated here, along with the caste system and other cultural traits

**1. Tkinter GUI Version**

A desktop interface to paste text and get a summary.

python

CopyEdit

import tkinter as tk

from tkinter import scrolledtext

from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

def summarize\_text():

input\_text = input\_box.get("1.0", tk.END).strip()

if input\_text:

summary = summarizer(input\_text, max\_length=130, min\_length=30, do\_sample=False)

output\_box.delete("1.0", tk.END)

output\_box.insert(tk.END, summary[0]['summary\_text'])

root = tk.Tk()

root.title("Article Summarizer")

root.geometry("800x600")

tk.Label(root, text="Enter Article Text:", font=("Arial", 12)).pack()

input\_box = scrolledtext.ScrolledText(root, height=15)

input\_box.pack(padx=10, pady=10, fill=tk.BOTH)

tk.Button(root, text="Summarize", command=summarize\_text, font=("Arial", 12)).pack()

tk.Label(root, text="Summary:", font=("Arial", 12)).pack()

output\_box = scrolledtext.ScrolledText(root, height=10)

output\_box.pack(padx=10, pady=10, fill=tk.BOTH)

root.mainloop()

**✅ 2. Flask Web App**

python

CopyEdit

from flask import Flask, request, render\_template\_string

from transformers import pipeline

app = Flask(\_\_name\_\_)

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

HTML\_TEMPLATE = '''

<!doctype html>

<title>Article Summarizer</title>

<h2>Paste Your Article Below:</h2>

<form method=post>

<textarea name=text rows=15 cols=100></textarea><br><br>

<input type=submit value=Summarize>

</form>

{% if summary %}

<h3>Summary:</h3>

<p>{{ summary }}</p>

{% endif %}

'''

@app.route('/', methods=['GET', 'POST'])

def summarize():

summary = ''

if request.method == 'POST':

text = request.form['text']

if text:

result = summarizer(text, max\_length=130, min\_length=30, do\_sample=False)

summary = result[0]['summary\_text']

return render\_template\_string(HTML\_TEMPLATE, summary=summary)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

🔧 To run:

bash

CopyEdit

python app.py

# Then open http://127.0.0.1:5000 in your browser.

**✅ 3. Streamlit App**

python

CopyEdit

import streamlit as st

from transformers import pipeline

st.set\_page\_config(page\_title="Article Summarizer", layout="wide")

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

st.title("📄 Article Summarizer using NLP")

st.markdown("Paste a long article below and get a concise summary!")

text = st.text\_area("✏️ Enter your article here", height=300)

if st.button("Summarize"):

if text:

with st.spinner("Summarizing..."):

result = summarizer(text, max\_length=130, min\_length=30, do\_sample=False)

st.subheader("📝 Summary:")

st.success(result[0]['summary\_text'])

else:

st.warning("Please enter some text to summarize.")

🔧 To run:

bash

CopyEdit

streamlit run streamlit\_app.py

**Part 1: Create a .exe Installer for the Tkinter App**

We'll turn the Tkinter-based article summarizer into a **Windows executable** using **PyInstaller**.

**🛠 Steps:**

**1. Save Your Tkinter Script as summarizer\_gui.py**

**2. Install PyInstaller**

bash

CopyEdit

pip install pyinstaller

**3. Build the Executable**

bash

CopyEdit

pyinstaller --noconsole --onefile --icon=icon.ico summarizer\_gui.py

Optional:

* --noconsole hides the terminal window.
* --icon=icon.ico adds a custom app icon (replace with your .ico file).

**4. Output**

You’ll find your .exe file in the dist/ folder:

bash

CopyEdit

dist/summarizer\_gui.exe

You can share this .exe with anyone to run the summarizer without Python installed.  
Let me know if you want a **dark-themed installer with branding**, and I’ll make a **Inno Setup** script for it.

**✅ Part 2: Deploy Flask & Streamlit to the Cloud**

**🌐 Option 1: Deploy Streamlit to Hugging Face Spaces (Best for Streamlit)**

**Steps:**

1. **Create a Hugging Face account** at [huggingface.co](https://huggingface.co).
2. Create a **New Space → Streamlit → Choose repo name**.
3. Upload these 2 files:
   * app.py (your Streamlit summarizer)
   * requirements.txt with:

txt

CopyEdit

transformers

torch

streamlit

1. Done! Hugging Face will build and host it automatically for free.

**🌐 Option 2: Deploy Flask to Render (Best for Flask Web App)**

**Steps:**

1. Sign up at <https://render.com>
2. Push your code (Flask app) to GitHub:

CopyEdit

app.py

requirements.txt

requirements.txt should include:

txt

CopyEdit

transformers

torch

Flask

1. On Render dashboard:
   * Create a new **Web Service**
   * Connect your repo
   * Set build command: pip install -r requirements.txt
   * Set start command: python app.py
2. Render deploys and gives you a public URL.

**1. FOLDER STRUCTURE (FULL PACKAGE)**

css

CopyEdit

article\_summarizer/

│

├── gui\_tkinter/

│ ├── summarizer\_gui.py

│ └── icon.ico # (optional) icon for .exe

│

├── flask\_webapp/

│ ├── app.py

│ └── requirements.txt

│

├── streamlit\_app/

│ ├── app.py

│ └── requirements.txt

│

├── inno\_setup\_installer/

│ └── summarizer\_installer.iss

│

└── README.md

**🖥️ 2. gui\_tkinter/summarizer\_gui.py**

python

CopyEdit

import tkinter as tk

from tkinter import scrolledtext

from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

def summarize\_text():

input\_text = input\_box.get("1.0", tk.END).strip()

if input\_text:

summary = summarizer(input\_text, max\_length=130, min\_length=30, do\_sample=False)

output\_box.delete("1.0", tk.END)

output\_box.insert(tk.END, summary[0]['summary\_text'])

root = tk.Tk()

root.title("Article Summarizer")

root.geometry("800x600")

tk.Label(root, text="Enter Article Text:", font=("Arial", 12)).pack()

input\_box = scrolledtext.ScrolledText(root, height=15)

input\_box.pack(padx=10, pady=10, fill=tk.BOTH)

tk.Button(root, text="Summarize", command=summarize\_text, font=("Arial", 12)).pack()

tk.Label(root, text="Summary:", font=("Arial", 12)).pack()

output\_box = scrolledtext.ScrolledText(root, height=10)

output\_box.pack(padx=10, pady=10, fill=tk.BOTH)

root.mainloop()

💡 To convert to .exe, use:

bash

CopyEdit

pyinstaller --noconsole --onefile --icon=icon.ico summarizer\_gui.py

**🌐 3. flask\_webapp/app.py**

python

CopyEdit

from flask import Flask, request, render\_template\_string

from transformers import pipeline

app = Flask(\_\_name\_\_)

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

HTML\_TEMPLATE = '''

<!doctype html>

<title>Article Summarizer</title>

<h2>Paste Your Article Below:</h2>

<form method=post>

<textarea name=text rows=15 cols=100></textarea><br><br>

<input type=submit value=Summarize>

</form>

{% if summary %}

<h3>Summary:</h3>

<p>{{ summary }}</p>

{% endif %}

'''

@app.route('/', methods=['GET', 'POST'])

def summarize():

summary = ''

if request.method == 'POST':

text = request.form['text']

if text:

result = summarizer(text, max\_length=130, min\_length=30, do\_sample=False)

summary = result[0]['summary\_text']

return render\_template\_string(HTML\_TEMPLATE, summary=summary)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**✅ flask\_webapp/requirements.txt**

nginx

CopyEdit

transformers

torch

Flask

**🎈 4. streamlit\_app/app.py**

python

CopyEdit

import streamlit as st

from transformers import pipeline

st.set\_page\_config(page\_title="Article Summarizer", layout="wide")

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

st.title("📄 Article Summarizer using NLP")

st.markdown("Paste a long article below and get a concise summary!")

text = st.text\_area("✏️ Enter your article here", height=300)

if st.button("Summarize"):

if text:

with st.spinner("Summarizing..."):

result = summarizer(text, max\_length=130, min\_length=30, do\_sample=False)

st.subheader("📝 Summary:")

st.success(result[0]['summary\_text'])

else:

st.warning("Please enter some text to summarize.")

**✅ streamlit\_app/requirements.txt**

nginx

CopyEdit

transformers

torch

streamlit

**🛠️ 5. inno\_setup\_installer/summarizer\_installer.iss**

iss

CopyEdit

[Setup]

AppName=Article Summarizer

AppVersion=1.0

DefaultDirName={pf}\ArticleSummarizer

OutputDir=output

OutputBaseFilename=ArticleSummarizerInstaller

Compression=lzma

SolidCompression=yes

[Files]

Source: "dist\summarizer\_gui.exe"; DestDir: "{app}"; Flags: ignoreversion

[Icons]

Name: "{group}\Article Summarizer"; Filename: "{app}\summarizer\_gui.exe"

👉 Run this .iss file in **Inno Setup Compiler** to create a branded installer.

**📄 README.md (Optional)**

md

CopyEdit

# 📰 Article Summarizer (3-in-1)

This project contains:

- A Tkinter GUI-based summarizer

- A Flask web app

- A Streamlit cloud-ready app

- .exe builder + installer script

## 💻 Run GUI

```bash

cd gui\_tkinter

python summarizer\_gui.py

**🌐 Run Flask**

bash

CopyEdit

cd flask\_webapp

pip install -r requirements.txt

python app.py

**📊 Run Streamlit**

bash

CopyEdit

cd streamlit\_app

pip install -r requirements.txt

streamlit run app.py

**🏗 Build .exe**

bash

CopyEdit

cd gui\_tkinter

pyinstaller --noconsole --onefile --icon=icon.ico summarizer\_gui.py

**🚀 Deploy Streamlit to Hugging Face**

Upload streamlit\_app/ folder to a new Hugging Face Space (type: Streamlit).

**🐳 Deploy Flask to Render**

Push flask\_webapp/ to GitHub → Deploy via [render.com](https://render.com)

**Automated Git Push Using Python + PyGithub**

Here’s a **ready-to-run Python script** that uses **PyGithub** to:

1. Create a GitHub repo (if not already created).
2. Push all files (for files under 25 MB) into the repo on a given branch.

You just need a **GitHub personal access token** with repo scope.

python

CopyEdit

# push\_to\_github.py

import os

from github import Github

import base64

# ===== CONFIG =====

GITHUB\_TOKEN = os.getenv("GITHUB\_TOKEN") # better to set in env var

REPO\_NAME = "article‑summarizer‑project"

REPO\_DESCRIPTION = "3‑in‑1 Article Summarizer: GUI, Flask, Streamlit"

BRANCH = "main"

LOCAL\_PROJECT\_DIR = "article\_summarizer"

# ===== SCRIPT =====

def main():

if not GITHUB\_TOKEN:

print("💡 Set your env var GITHUB\_TOKEN before running.")

return

g = Github(GITHUB\_TOKEN)

user = g.get\_user()

# create repo if not exists

try:

repo = user.get\_repo(REPO\_NAME)

print(f"✅ Repo exists: {REPO\_NAME}")

except:

repo = user.create\_repo(REPO\_NAME, description=REPO\_DESCRIPTION, private=False)

print(f"🚀 Created repo: {REPO\_NAME}")

# Push each file from local folder

for root, \_, files in os.walk(LOCAL\_PROJECT\_DIR):

for fname in files:

local\_path = os.path.join(root, fname)

rel\_path = os.path.relpath(local\_path, LOCAL\_PROJECT\_DIR)

with open(local\_path, "rb") as f:

content = f.read()

try:

# check if exists

contents = repo.get\_contents(rel\_path, ref=BRANCH)

repo.update\_file(

path=rel\_path,

message=f"Update {rel\_path}",

content=content,

sha=contents.sha,

branch=BRANCH

)

print(f"🔄 Updated: {rel\_path}")

except Exception:

repo.create\_file(

path=rel\_path,

message=f"Add {rel\_path}",

content=content,

branch=BRANCH

)

print(f"➕ Created: {rel\_path}")

print("🎉 All done! Visit: https://github.com/" + user.login + "/" + REPO\_NAME)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**✅ How to use:**

1. Install:

bash

CopyEdit

pip install PyGithub

1. Ensure your folder structure is at article\_summarizer/ (like I described earlier).
2. Set your GitHub API token:

bash

CopyEdit

export GITHUB\_TOKEN="YOUR\_TOKEN\_HERE"

1. Run:

bash

CopyEdit

python push\_to\_github.py

It will tell you:

* If the repo exists or gets created,
* Which files are created or updated,
* And finally give you the URL to visit.

This uses **PyGithub**, which is ideal for managing GitHub content programmatically—no git commands needed! [Stack Overflow](https://stackoverflow.com/questions/75327473/how-do-you-push-files-to-a-github-repo-in-python?utm_source=chatgpt.com)[Gist+6GitHub+6GitHub+6](https://github.com/PyGithub/PyGithub?utm_source=chatgpt.com)[techgeekbuzz.com](https://www.techgeekbuzz.com/blog/how-to-use-github-api-in-python/?utm_source=chatgpt.com)

**🧾 Summary of Options**

| **Option** | **Description** |
| --- | --- |
| **Shell script** (git add . && git commit -m … && git push) | Quick and simple for manual CLI pushing. [Gist+3Blog+3DEV Community+3](https://kevquirk.com/blog/git-commit-and-push-script?utm_source=chatgpt.com) |
| **Python + GitHub API (PyGithub)** | Full automation via script and GitHub API—works even on systems without git. |
| **CI/CD (e.g., GitHub Actions, Travis CI)** | Automatically push on schedule or trigger; ideal for automation. |

**💌 What’s next?**

After running the script, you'll have a link like:

arduino

CopyEdit

https://github.com/yourusername/article‑summarizer‑project