

```

#I am Installing necessary dependencies like playwright and installing chromadb and generative ai
!pip install -q playwright
!playwright install chromium

!pip install -q google-generativeai
!pip install -q chromadb ipywidgets

/usr/lib/python3.11/pathlib.py:542: RuntimeWarning: coroutine 'scrape_chapter' was never awaited
  self._str = self._format_parsed_parts(self._drv, self._root,
RuntimeWarning: Enable tracemalloc to get the object allocation traceback

#configuring gemini api
import google.generativeai as genai

genai.configure(api_key="AIzaSyCYlHs9qzR9uSjTRR6msQ70vFsiovGCe4g")#this is my gemini api key
model = genai.GenerativeModel("gemini-1.5-flash-latest")

def call_gemini(prompt):
    response = model.generate_content(prompt)
    return response.text

#Scrape Chapter and Screenshot from the url
import asyncio
from playwright.async_api import async_playwright
import nest_asyncio

async def scrape_chapter(url, screenshot_path="chapter1.png"):
    async with async_playwright() as p:
        browser = await p.chromium.launch()
        page = await browser.new_page()
        await page.goto(url)
        text = await page.inner_text("body")
        await page.screenshot(path=screenshot_path, full_page=True)
        await browser.close()
        return text

url = "https://en.wikisource.org/wiki/The_Gates_of_Morning/Book_1/Chapter_1"
nest_asyncio.apply() # Apply nest_asyncio to allow asyncio.run() in Colab
chapter_text = asyncio.run(scrape_chapter(url))
print(chapter_text[:1000]) # preview

Jump to content
Main menu
Search
Donate
Create account
Log in
Personal tools
The Gates of Morning/Book 1/Chapter 1
Add languages
Page
Source
Discussion
Read
Edit
View history
Tools
Appearance hide
Text
Small
Standard
Large
Width
Standard
Wide
Download
< The Gates of Morning | Book 1

←
Book 1
The Gates of Morning/Book 1
by Henry De Vere Stacpoole
Book 1. Chapter 1
Chapter 2
→

"The Gates of Morning"

CHAPTER I

THE CANOE BUILDER

```

```

D
ICK standing on a ledge of coral cast his eyes to the South.

Behind him the breakers of the outer sea thundered and the spindrift scattered on the wind; before him stretched an ocean calm as a
Clipped by its forty-mile ring of coral this great pond was a sea in itself, a sea of storm in heavy winds, a lake of azure, in light
Women, children, youths, all the tribe to be seen busy along the beach in the blazing

# AI Writer & Reviewer (Using Gemini)
def ai_writer(text):
    prompt = f"Rewrite this creatively and engagingly:\n\n{text[:10000]}"
    return call_gemini(prompt)

def ai_reviewer(text):
    prompt = f"Review this text and improve clarity, grammar, and engagement:\n\n{text[:10000]}"
    return call_gemini(prompt)

spun_text = ai_writer(chapter_text)
reviewed_text = ai_reviewer(spun_text)

#Human-in-the-Loop Editing
from IPython.display import display
import ipywidgets as widgets

editor = widgets.Textarea(
    value=reviewed_text,
    placeholder="Edit the reviewed version here...",
    layout=widgets.Layout(width='100%', height='300px')
)
display(editor)

```

```

## The Gates of Morning: Chapter 1 - The Canoe Builder

**A Coral Throne, a Heart Aflame:**

Dick perched on a coral ledge, gazing south. Behind him, the ocean raged—a tempest of whitecaps and wind-whipped spray. Before him, Karolin's lagoon shimmered, a tranquil expanse of azure dotted with graceful fishing gulls. This breathtaking lagoon, protected by forty miles of coral reef, was his—his since yesterday, when he first set foot on its shores.

His new people—women, children, and youths—toiled beneath the relentless sun, their lives a vibrant tapestry woven from fishing nets, playful shouts, and the rhythmic tapping of paraka patch creation. His were the canoes beached on the sand; his, the empty houses that once echoed with Karolin's warriors' cries.

But a shadow crossed his brow as his gaze shifted from the lagoon to the deserted canoe houses. He turned, facing the relentless fury of the northern sea, beyond which lay Palm Tree—a breathtaking paradise... haunted by devils.

Little Tari, son of Le Taioi the net-maker, watched from the coral. Untouched by life's harsh realities, Tari only knew that war had swept away Karolin's men, leaving women and children defenseless. Then, yesterday, a miracle: a stranger, arriving in a strange boat alongside Katafa, the girl lost at sea years ago, had appeared. The women whispered—a god-sent chief!

Unaware of such whispers, Tari, emboldened by the stranger's presence, blurted out a simple question: "Taori," he asked, "who *are*

```

```

final_edited_text = editor.value
print(final_edited_text)

```

```

## The Gates of Morning: Chapter 1 - The Canoe Builder

**A Coral Throne, a Heart Aflame:**

Dick perched on a coral ledge, gazing south. Behind him, the ocean raged—a tempest of whitecaps and wind-whipped spray. Before him
His new people—women, children, and youths—toiled beneath the relentless sun, their lives a vibrant tapestry woven from fishing n
But a shadow crossed his brow as his gaze shifted from the lagoon to the deserted canoe houses. He turned, facing the relentless
Little Tari, son of Le Taioi the net-maker, watched from the coral. Untouched by life's harsh realities, Tari only knew that war
Unaware of such whispers, Tari, emboldened by the stranger's presence, blurted out a simple question: "Taori," he asked, "who *are*
Could Dick have answered truthfully? Could Tari have understood the epic saga of his life? A life begun as a boy abandoned on Ma
No. The name "Dick" was lost to the mists of his forgotten past, swallowed by Marua's sun and storms. Only the memory of Katafa a
He looked down at the inquisitive child. "I am Taori, Tari tatu. Why do you ask?"

"I don't know," Tari replied, his attention already diverted by children hauling in a net. "I ask as I breathe..."

Dick's gaze returned to the north. Tari's question stirred a dormant memory: the schooner, the Melanesians, the desperate flight
Palm Tree posed no immediate threat; its inhabitants lacked canoes. But they *would* build them. They *would* see the shimmering

```

Dick, though a kanaka, possessed the strategic mind of a white man. He could anticipate, plan, and envision. That morning, he'd s

War canoes... defense or offense? The question remained unanswered.

A hand touched his shoulder. Katafa. A strand of dark hair, unbound from its vine, danced in the breeze like an eagle's feather.

"Look," she whispered.

Through a mirage, Palm Tree, usually hidden, emerged from the sea—a ghostly apparition, beautiful and terrifying. They stood sile

Dick's nostrils flared, his eyes burning with hate. They had threatened him; they had threatened *Katafa*. And the threat remaine

One day, they would come. Palm Tree was a warning. They would build canoes, see the lagoon's light, and they would come. Drawn by

Then, as swiftly as it appeared, the vision of Palm Tree vanished. Katafa pointed to a canoe approaching from the south—the canoe

(To be continued...)

Changes Made:

- * **Stronger Verbs:** Replaced weaker verbs (e.g., "roared its defiance" with "raged") for a more impactful narrative.
- * **Conciseness:** Removed redundant phrases and wording.
- * **Sentence Structure:** Varied sentence length and structure for better rhythm and flow.
- * **Show, Don't Tell:** Focused more on descriptive language to evoke imagery and emotion rather than simply stating facts.
- * **Improved Pacing:** Adjusted the pacing to create more suspense and anticipation.
- * **Clarity:** Clarified ambiguous phrases and improved transitions between paragraphs.
- * **Emphasis:** Used italics strategically for emphasis.
- * **Word Choice:** Replaced some words with more evocative synonyms (e.g., "reef" instead of "protective coral").

#Version Control with ChromaDB

```
import chromadb
from chromadb.config import Settings

client = chromadb.Client(Settings(anonymized_telemetry=False))
collection = client.get_or_create_collection(name="book_versions")

def save_version(book, chapter, version_id, content, role):
    doc_id = f"{book}_{chapter}_v{version_id}_{role}"
    metadata = {
        "book": book,
        "chapter": chapter,
        "version": version_id,
        "role": role
    }
    collection.add(documents=[content], metadatas=[metadata], ids=[doc_id])

save_version("The Gates of Morning", "Book1_Chapter1", "1", chapter_text, "original")
save_version("The Gates of Morning", "Book1_Chapter1", "2", spun_text, "writer")
save_version("The Gates of Morning", "Book1_Chapter1", "3", reviewed_text, "reviewer")
save_version("The Gates of Morning", "Book1_Chapter1", "4", final_edited_text, "editor")
```

#Semantic Search with ChromaDB

```
def semantic_search(query, top_k=5):
    results = collection.query(query_texts=[query], n_results=top_k)
    return results

results = semantic_search("Sea voyage and survival")

for doc, meta in zip(results["documents"][0], results["metadatas"][0]):
    print(f"Version {meta['version']} ({meta['role']}): \n{doc[:300]}")
    print("---")
```

```
Version 2 (writer):
## The Gates of Morning: A Coral Kingdom's Dawn

**(Chapter 1: The Canoe Builder)**

The sun blazed, a molten coin sinking into the turquoise expanse of Karolin lagoon. Dick, perched on a coral ledge, surveyed his ki
---
Version 3 (reviewer):
## The Gates of Morning: A Coral Kingdom's Dawn

**Chapter 1: The Canoe Builder**

The sun, a molten coin sinking into Karolin Lagoon's turquoise embrace, cast long shadows across the coral. From his perch, Dick si
---
Version 4 (editor):
```

```

## The Gates of Morning: A Coral Kingdom's Dawn

**Chapter 1: The Canoe Builder**

The sun, a molten coin sinking into Karolin Lagoon's turquoise embrace, cast long shadows across the coral. From his perch, Dick st
---
Version 1 (original):
Jump to content
Main menu
Search
Donate
Create account
Log in
Personal tools
The Gates of Morning/Book 1/Chapter 1
Add languages
Page
Source
Discussion
Read
Edit
View history
Tools
Appearance hide
Text
Small
Standard
Large
Width
Standard
Wide
Download
< The Gates of Morning | Book 1

←
Book 1
T
---

# RL-Guided Retrieval with Feedback
import random

q_table = {}

def update_q_table(state, action, reward, alpha=0.1, gamma=0.9):
    old_value = q_table.get((state, action), 0.0)
    future_rewards = [q_table.get((state, a), 0.0) for a in ['original', 'writer', 'reviewer', 'editor']]
    new_value = old_value + alpha * (reward + gamma * max(future_rewards) - old_value)
    q_table[(state, action)] = new_value

#RL Search Function
def rl_guided_search(state, query, top_k=5):
    results = semantic_search(query, top_k=top_k)
    role_to_doc = {meta['role']: (doc, meta)
                    for doc, meta in zip(results["documents"][0], results["metadatas"][0])}
    available_roles = list(role_to_doc.keys())

    if not available_roles:
        return None

    if random.random() < 0.2: # Exploration
        chosen_role = random.choice(available_roles)
    else: # Exploitation
        chosen_role = max(available_roles, key=lambda r: q_table.get((state, r), 0.0))

    return role_to_doc[chosen_role], chosen_role

#User Feedback Loop:
def rl_episode(state, query):
    result, role = rl_guided_search(state, query)
    if not result:
        print("No versions found.")
        return

    content, meta = result
    print(f"\n 📄 Preview ({role}): \n", content[:500])

    try:
        reward = int(input(f"\nRate the '{role}' version (0-10): "))
    except:

```

```
reward = 5 # default

update_q_table(state, role, reward)
print(f"✅ Updated Q-value for ({state}, {role}): {q_table[(state, role)]}")
```

Start coding or [generate](#) with AI.