Experiment 5:

Use word embedding's to create meaningful sentences for creative tasks.

(i) Retrieve similar words for a seed word.

(ii) Create a sentence or story using these words as a starting point. Write a program that:

(a) Takes a seed word.

(b) Generates similar words.

(c)Constructs a short paragraph using these words.

AIM: To build a user friendly application that takes a seed word as input, retrieves semantically or creatively similar words using Googles Gemini LLM via LangChain, and generates creative paragraph variations based on those words.

Introduction:

Large Language Models (LLMs)

LLMs are advanced AI systems trained on vast amounts of text data to generate human like text. They understand context, semantics, and syntax, allowing them to write essays, generate code, answer questions, and even compose poetry.

Gemini's Flash Model

Google's Gemini Flash is a lightweight, fast, and efficient version of the Gemini LLM family. It is optimized for quick response times and is ideal for real time applications such as chatbots, text generation, and summarization. In this lab, we use gemini-2.0-flash-exp, an experimental fast-response model.

LangChain Library

LangChain is a framework that simplifies the development of applications powered by language models. It provides tools to connect to different LLMs (like OpenAI, Cohere, Gemini, etc.), manage conversations, chain together prompts, and integrate memory, tools, and agents. We use LangChain to send prompts to Gemini and receive structured responses.

Gradio Web Interface

Gradio is an open-source Python library that allows us to quickly create web apps to showcase machine learning models or functions. It provides components like text boxes, buttons, and file downloaders so users can interact with the model in a browser with no coding required.

Large Language Models (LLMs) such as Google's Gemini can generate human-like responses and creative content. In this lab, we use the LangChain library to interact with Gemini's Flash model and create a simple Gradio web interface for creative paragraph generation. The application takes a single word (the seed word) and builds imaginative content based on related terms.

Objectives of the Program

1. Understand how to integrate and interact with Google’s Gemini LLM in a Python application using the LangChain framework.
2. Retrieve semantically or creatively related words for a given seed word using the Gemini language model.
3. Generate multiple creative paragraph variations that meaningfully use the seed word and its related words.
4. Design and implement a user friendly browser based interface using the Gradio library to showcase the application.

Step 1: Install Required Libraries

```
!pip install -q langchain-google-genai gradio
```

Step 2: Imports, Configure and Gemini LLM setup (Initialize Gemini LLM)

```
# Imports and Gemini LLM setup

from langchain_google_genai import ChatGoogleGenerativeAI
from langchain_core.messages import HumanMessage
import gradio as gr
import getpass
import io



# Get Google API key securely
GOOGLE_API_KEY = getpass.getpass("Enter your Google API key: ")
```

```
# Initialize Gemini 2.0 LLM
llm = ChatGoogleGenerativeAI(
    model="gemini-2.0-flash-exp",
    temperature=0.8,
    api_key=GOOGLE_API_KEY,
    max_tokens=512,
    timeout=30,
    max_retries=2,
)

print("Gemini LLM is ready.")
```

Output: *Enter your Google API key: ·····*

Gemini LLM is ready.

Step 3: Retrieve Similar Words from Gemini LLM for the Given Seed Words

```
# Get similar words from Gemini
def get_similar_words(seed):
    prompt =(
        f"Give me 5 English words that are semantically or creatively similar to '{seed}'. "
        f"Return the words as a comma separated list without numbers or explanations."
    )
```

```
    response = llm.invoke([HumanMessage(content=prompt)])
    return [word.strip() for word in response.content.split(',') if word.strip()]
```

Step 4: Generate creative paragraphs **using** seed words

```
def create_paragraph(seed, words):
    word_list = ', '.join(words)
    prompt = (
          f"Write a short, creative paragraph using the words '{seed}' and the following related words: "
          f"{word_list}. The paragraph should be imaginative and meaningful."
    )
    response = llm.invoke([HumanMessage(content=prompt)])
    return response.content
```

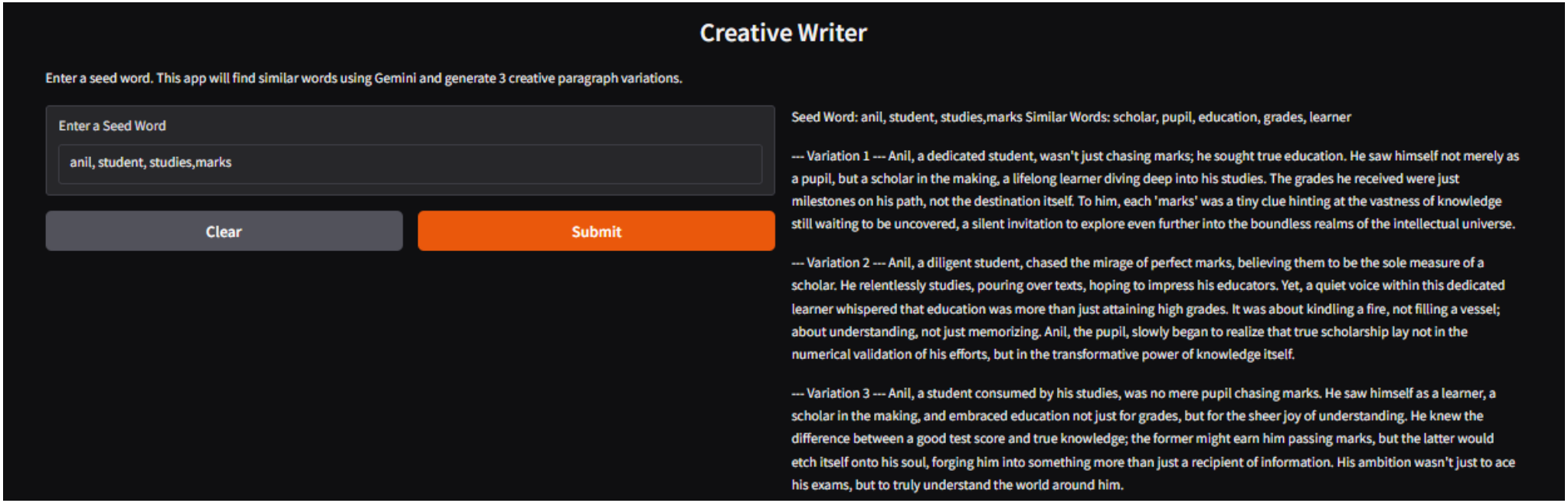
Step 5: Generate and Format Paragraph Variations Using the Seed Words and Similar Words

```
def generate_paragraphs(seed_word):
    try:
        seed_word = seed_word.strip()
        if not seed_word:
            return "Please enter a valid seed word."
        similar_words = get_similar_words(seed_word)
        if len(similar_words) < 3:
            return "Could not find similar words. Try a different seed word."

        output_text = f"Seed Word: {seed_word}\nSimilar Words: {', '.join(similar_words)}\n\n"

        for i in range(1, 4):
            paragraph=create_paragraph(seed_word, similar_words) or f"(Variation{i})Could not generate paragraph."
            output_text += f"--- Variation {i} ---\n{paragraph.strip()}\n\n"

        return output_text

    except Exception as e:
        return f"Error: {str(e)}"
```

Step 6: Build and Launch the Gradio User Interface for Paragraph Generation

```
gr.Interface(
    fn=generate_paragraphs,
    inputs=gr.Textbox(label="Enter a Seed Word"),
    outputs=gr.Markdown(label="Generated Paragraphs"),
    title="Creative Writer",
    description="Enter a seed word. This app will find similar words using Gemini and generate 3 creative paragraph variations.",
    theme="default",
).launch(debug=False)
```

Output:


Creative Writer
Enter a seed word. This app will find similar words using Gemini and generate 3 creative paragraph variations.
Enter a Seed Word
anil, student, studies,marks
Clear
Submit
Seed Word: anil, student, studies,marks Similar Words: scholar, pupil, education, grades, learner
--- Variation 1 --- Anil, a dedicated student, wasn't just chasing marks; he sought true education. He saw himself not merely as a pupil, but a scholar in the making, a lifelong learner diving deep into his studies. The grades he received were just milestones on his path, not the destination itself. To him, each 'marks' was a tiny clue hinting at the vastness of knowledge still waiting to be uncovered, a silent invitation to explore even further into the boundless realms of the intellectual universe.
--- Variation 2 --- Anil, a diligent student, chased the mirage of perfect marks, believing them to be the sole measure of a scholar. He relentlessly studies, pouring over texts, hoping to impress his educators. Yet, a quiet voice within this dedicated learner whispered that education was more than just attaining high grades. It was about kindling a fire, not filling a vessel; about understanding, not just memorizing. Anil, the pupil, slowly began to realize that true scholarship lay not in the numerical validation of his efforts, but in the transformative power of knowledge itself.
--- Variation 3 --- Anil, a student consumed by his studies, was no mere pupil chasing marks. He saw himself as a learner, a scholar in the making, and embraced education not just for grades, but for the sheer joy of understanding. He knew the difference between a good test score and true knowledge; the former might earn him passing marks, but the latter would etch itself onto his soul, forging him into something more than just a recipient of information. His ambition wasn't just to ace his exams, but to truly understand the world around him.