

TEXT COMPLETION GENERATOR

Phase 2: Project Execution and Demonstration

1. Project Title:

The title of the project is “**Text completion Generator**” and **Text completion** is a key task in Natural Language Processing (NLP), where the goal is to predict the continuation of a given text based on its preceding context. With the evolution of **Generative Artificial Intelligence (Gen AI)**, this task has seen significant advancements, making it possible for machines to generate human-like, contextually accurate, and grammatically correct text.

2. Objective Recap:

The objective of this project is to build a **Text Completion** system using **Generative AI** models. The system takes an input sentence and generates one or more plausible next sentences based on the context, using state-of-the-art NLP models.

In the realm of Natural Language Processing (NLP), the ability to understand and generate coherent text is a fundamental challenge. This capability is crucial for developing intelligent applications such as chatbots, content generators, and summarization tools. A key aspect of this challenge is predicting the next sentence or phrase based on a given context, enabling machines to produce human-like text that is contextually relevant and meaningful.

Recent advancements in Generative AI, particularly the development of transformer-based models like GPT-2 and GPT-3, have significantly enhanced the performance of text generation tasks. These models are trained on vast corpora of text data, allowing them to capture intricate patterns in language and generate coherent continuations for a given input. Such capabilities have opened new avenues for applications that require dynamic and context-aware text generation.

This project aims to develop a Generative AI-based system that can predict or suggest the most appropriate next sentence for a given input. By leveraging pre-trained transformer models and integrating them into an interactive web application using Streamlit, the system provides users with an intuitive interface to experience real-time text completion.

3. Technologies Used:

- Python
- HuggingFace Transformers
- Streamlit (for web interface)
- Google Colab / Jupyter Notebook
- Pre-trained GPT-2 Model

4. Full Code Implementation:

Step 1: Install Required Libraries

```
pip install transformers streamlit
```

Step 2: Import Required Libraries

```
from transformers import pipeline, set_seed  
import streamlit as st
```

Step 3: Load the Pretrained GPT-2 Model

```
generator = pipeline('text-generation', model='gpt2')  
set_seed(42)
```

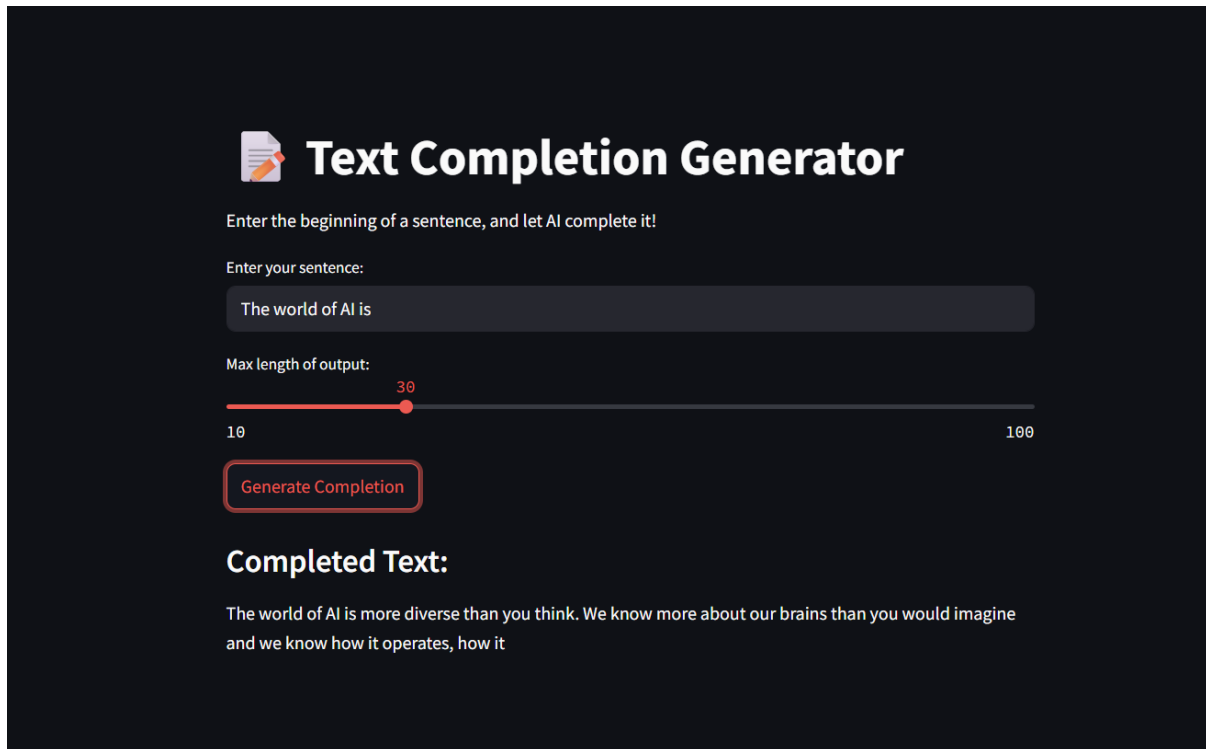
Step 4: Build Streamlit Interface


```
st.title("Next Sentence Prediction using Generative AI")  
st.write("Enter a sentence to predict the most likely next sentence.")  
  
input_text = st.text_input("Enter your sentence here:")  
  
if input_text:  
    st.subheader("Generated Sentences:")  
    outputs = generator(input_text, max_length=50, num_return_sequences=3)  
    for i, output in enumerate(outputs):  
        st.write(f"{i+1}. {output['generated_text']}")
```

Step 5: Run the Streamlit App

```
streamlit run app.py
```

5. Output Screenshots:



 **Text Completion Generator**

Enter the beginning of a sentence, and let AI complete it!

Enter your sentence:

The world of AI is

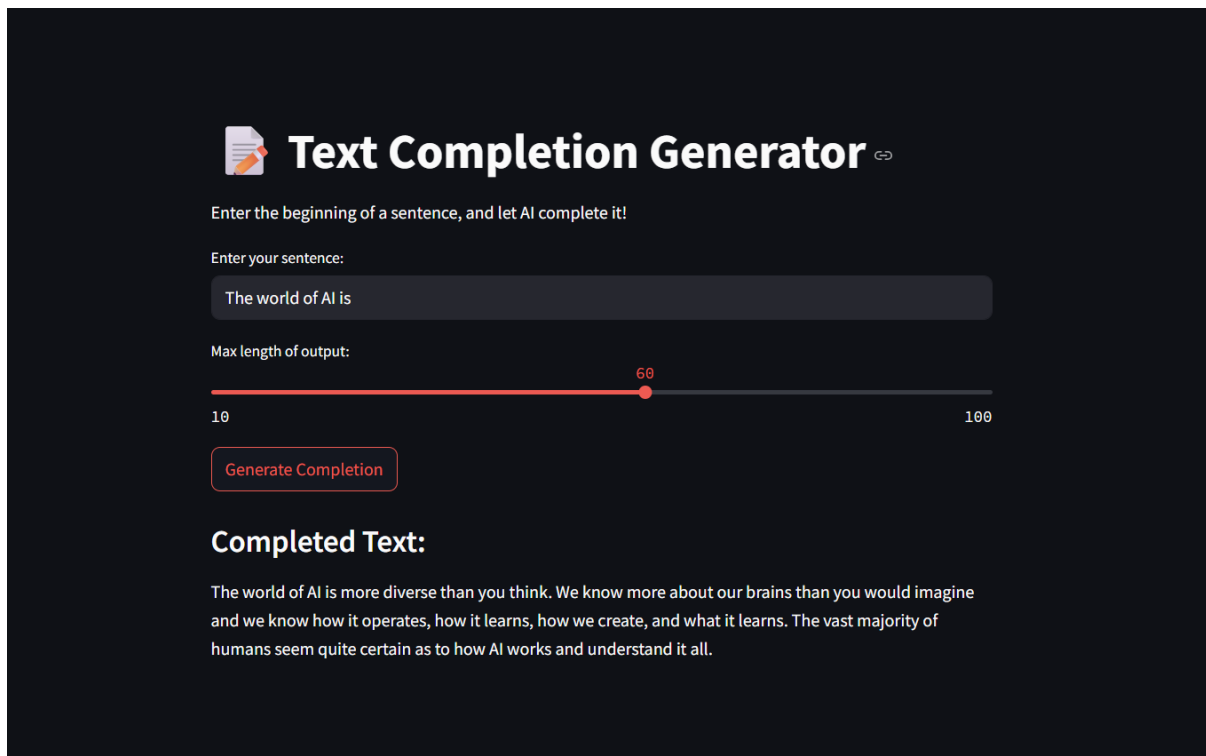
Max length of output:


10 30 100

Generate Completion

Completed Text:

The world of AI is more diverse than you think. We know more about our brains than you would imagine and we know how it operates, how it



 **Text Completion Generator**

Enter the beginning of a sentence, and let AI complete it!

Enter your sentence:

The world of AI is

Max length of output:

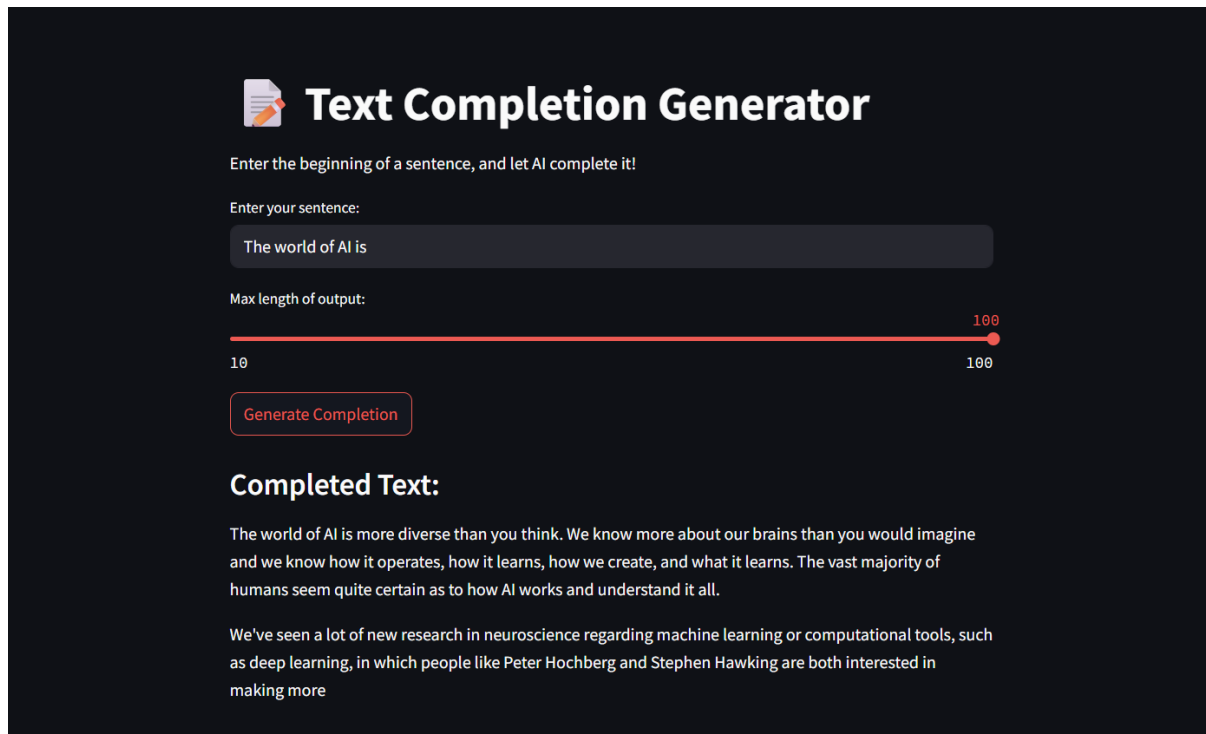
10 60 100

Generate Completion

Completed Text:

The world of AI is more diverse than you think. We know more about our brains than you would imagine and we know how it operates, how it learns, how we create, and what it learns. The vast majority of humans seem quite certain as to how AI works and understand it all.

Generated Text



The screenshot shows a web application titled "Text Completion Generator". It has a dark background with white and red text. The interface includes a text input field with the placeholder "Enter your sentence:" and the text "The world of AI is". Below the input field is a slider for "Max length of output:" ranging from 10 to 100. A red button labeled "Generate Completion" is positioned below the slider. The output section, titled "Completed Text:", displays the generated text: "The world of AI is more diverse than you think. We know more about our brains than you would imagine and we know how it operates, how it learns, how we create, and what it learns. The vast majority of humans seem quite certain as to how AI works and understand it all. We've seen a lot of new research in neuroscience regarding machine learning or computational tools, such as deep learning, in which people like Peter Hochberg and Stephen Hawking are both interested in making more".

6. Conclusion:

This project successfully implements a **Text Completion** system using the **GPT-2 language model**, highlighting the practical application of **Generative AI** in the domain of **Natural Language Processing (NLP)**. By leveraging the power of pre-trained transformer models, the system can generate meaningful and contextually relevant continuations to a given input sentence or paragraph.

The application provides an intuitive interface through **Streamlit**, allowing users to interact with the model in real-time. It showcases how AI can assist in **smart content creation**, enable **context-aware chatbot conversations**, and serve as a foundational tool for more advanced NLP applications such as **story generation**, **automated summarization**, and **intelligent writing support**.

This project not only demonstrates the capabilities of generative models like GPT-2 but also sets the stage for future enhancements by integrating larger models, incorporating user feedback for fine-tuning, and extending it to domain-specific use cases such as education, customer support, or creative writing.

7. References:

- **HuggingFace** Transformers Documentation
- **OpenAI** GPT-2 Research and API
- **Streamlit** Documentation

- **LangChain Tutorial: Build an LLM-powered App** – A tutorial on creating LLM-powered applications with Streamlit
- **BERT Implementation: A Step-by-Step Guide** – A comprehensive guide to implementing BERT for various NLP tasks
- **Auto-Completion Style Text Generation with GPT-2** – A tutorial on implementing auto-completion using GPT-2. [MachineLearningMastery.com](https://machinelearningmastery.com/text-generation-with-gpt-2/)
- **OpenAI's GPT-2: Building a Text Generator in Python** – A guide to building a GPT-2-based text generator. [Analytics Vidhya](https://analyticsvidhya.com/gpt-2-text-generation/)
- **Text Generation with GPT-2 Model** – A guide to implementing text generation using GPT-2. [MachineLearningMastery.com](https://machinelearningmastery.com/text-generation-with-gpt-2/)
- **Build a Text Summarization App Using Streamlit** – A guide to creating a text summarization application with Streamlit. [Python Simplified](https://python-simplified.com/build-a-text-summarization-app-using-streamlit/)
- **LangChain Tutorial: Build an Ask the Data App** – A tutorial on creating data-driven applications with LangChain and Streamlit. [Streamlit](https://streamlit.io/)