

Development and Evaluation of a Text Generation Application

1. Introduction

The rise of generative AI has transformed various domains, including content creation, personalized recommendations, and educational tools. This project focuses on developing a text generation application aimed at promoting Sustainable Development Goals (SDGs) using an open-source generative AI model.

The primary objective is to create an application that generates coherent, creative, and relevant text on SDG topics using models like BLOOM or GPT-2.

2. Tools and Libraries

Transformers Library: The `transformers` library by Hugging Face provides access to a wide range of pre-trained models, making it easier to implement state-of-the-art NLP tasks, including text generation.

Torch Library: `torch` is essential for handling the deep learning operations behind the scenes. It provides support for tensor computation and optimization.

Model Selection: The `bloom-560m` model was chosen for this task. It balances computational efficiency with performance, making it suitable for generating meaningful text within the constraints of available resources.

3. Implementation

3.1 Dataset for Text Generation:

A dataset of prompts related to Sustainable Development Goals (SDGs) is used to evaluate the text generation application.

Example Prompts:

"How can we achieve affordable and clean energy?"

"Discuss the impact of quality education on poverty reduction."

"What are the steps to ensure clean water and sanitation for all?"

"How does climate action contribute to sustainable cities?"

"Explain the role of gender equality in economic growth."

3.2 Model and Tokenizer Initialization:

```
model_name = "bigscience/bloom-560m"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)
```

The model and tokenizer are initialized with `bloom-560m`, ensuring that the application is equipped to process and generate text efficiently.

3.3 Text Generation Function:

```
def generate_text(prompt, max_length=100):
    inputs = tokenizer(prompt, return_tensors="pt")
    outputs = model.generate(inputs["input_ids"], max_length=max_length,
num_return_sequences=1)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)
```

This function takes a user prompt and produces a text continuation, which is both relevant to the input and contextually meaningful.

3.4 User Interface:

```
def main():
    print("Welcome to the SDG Text Generation App!")
    prompt = input("Enter your prompt related to SDGs: ")
    generated_text = generate_text(prompt)
    print("\nGenerated Text:\n", generated_text)
```

The simple command-line interface allows users to input a prompt and receive generated text, making the application interactive and user-friendly.

4. Evaluation

4.1 Dataset for Embeddings Comparison:

Two datasets, each containing sentences related to SDGs, are used to generate embeddings using models like Word2Vec, GloVe, and BERT.

Dataset 1:

"This is a sample sentence about climate action."
"Another example of a sentence related to quality education."
"Ensuring clean water and sanitation is crucial for health."
"Gender equality leads to economic growth and stability."
"Affordable and clean energy is key to sustainable development."

Dataset 2:

"Climate action is necessary for preserving the environment."

"Quality education is essential for eradicating poverty."

"Access to clean water is a fundamental human right."

"Empowering women boosts the economy and promotes equality."

"Renewable energy sources are vital for a sustainable future."

4.2 Coherence and Creativity

The model's ability to generate coherent and creative text is evaluated through various prompts. For instance, prompts related to 'Climate Action' or 'Quality Education' produce texts that are logically consistent and offer novel ideas.

4.3 Relevance

The generated content is aligned with the SDG topics. For example, a prompt about 'Clean Water and Sanitation' yields text focused on water conservation and access to clean drinking water.

4.4 Grammatical Correctness

The text generated is generally grammatically correct, with few instances of awkward phrasing, especially with shorter prompts.

5. Conclusion**5.1 Summary**

The text generation application successfully generates relevant and creative text on SDG topics. While the BLOOM-560m model is effective, further improvements could involve using larger models or fine-tuning for specific SDG-related datasets.

5.2 Future Improvements

Future work could include a more sophisticated user interface, integration with larger models for enhanced performance, and additional evaluation metrics to measure text quality.